International Islamic University Chittagong

Department of Computer Science and Engineering

Thesis on

PolypNet: An Attention Based Shallow Deep Learning Model for Colorectal Polyp Image Classification

Supervised by

Md. Khaliluzzaman
Associate Professor
Dept. of CSE, IIUC

Submitted by

Md. Saiful Islam
Matric No.: MC-201104R

Approval of the Supervisor

Declaration

I hereby declare that the work has been done by ourselves and no portion of the work contained in this thesis has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Md. Saiful Islam

November 7, 2025

Dedication

This thesis is dedicated to

My Thesis Supervisor

Acknowledgment

Education, along with the process of gaining knowledge and mastery of the subject, is a continuous journey. It is an appropriate blend of mindset, learned skills, experience, and knowledge obtained from various resources.

This thesis would not have been possible without the support of many people. First and foremost, I would like to express my deepest gratitude to the Almighty Allah, without whose support I would not have been able to complete the significant task of preparing this thesis within the scheduled time.

Additionally, I would like to express my heartfelt gratitude to Associate Professor Md. Khaliluzzaman for his invaluable guidance, which made the meaningful completion of this thesis possible. His new ideas and directions helped me navigate various areas of image compression techniques that were unfamiliar to me. I am also thankful for assigning me this intriguing thesis and for his valuable suggestions and encouragement throughout my research.

Finally, I would like to thank all my honorable teachers who have patiently assisted me throughout my thesis work.

Md. Saiful Islam (MC-201104R)

M.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering

IIUC, Chattogram

Abstract

Colon polyps are small, pre-cancerous growths in the colon that can indicate the presence of colorectal cancer (CRC), a disease that significantly impacts the well-being of individuals globally. Colonoscopy is a medical diagnostic technique used to find colon polyps. However, the manual examination and classification of polyps can be time-consuming, repetitive, and prone to human errors. Automation can improve the effectiveness of polyp classification in colonoscopy images. More specialized methodologies are needed to classify polyps identified after colonoscopy. Moreover, in small datasets, deep learning models often show overfitting problems. In this regard, the presented research introduces PolypNet, a shallow CNN-based deep-learning (DL) method combined with a self-attention mechanism for automatic colorectal polyp classification and overcoming the overfitting problem. To evaluate the performance of the model, several assessments are conducted using the highly acknowledged open-source benchmark dataset Kvasirv1, which has a total of 4000 images categorized into eight classes. The study compares four CNN based transfer-learning models for polyp classification, including VGG16, ResNet50, DenseNetv3, and MobileNetv3. The proposed model achieves an accuracy of 0.86, nearly identical to ResNet50 while outperforming VGG16, DenseNetv3, and MobileNetv3. The ablation study shows the significance of the proposed model.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ADR: Adenoma Detection Rate

AI: Artificial Intelligence

ANN: Artificial Neural Network

CAD: Computer-Aided Diagnosis

CNN: Convolutional Neural Network

ConvNet: Convolutional Network

CRC: Colorectal Cancer

DL: Deep Learning

FC: Fully Connected (Layer)

FN: False Negative

FP: False Positive

GAN: Generative Adversarial Network

GPU: Graphics Processing Unit

ILSVRC: ImageNet Large Scale Visual Recognition Challenge

mAP: mean Average Precision

MSE: Mean Square Error

NAS: Neural Architecture Search

ReLU: Rectified Linear Unit

RNN: Recurrent Neural Network

SGD: Stochastic Gradient Descent

TN: True Negative

TP: True Positive

ViT: Vision Transformer

YOLO: You Only Look Once

1. CHAPTER 1: INTRODUCTION

1.1 Overview

"Polyps" is the clinical word for any mucosal protrusion. A collection of cells that develop on the colon's lining is called a colon polyp. Although the majority of colon polyps are benign, some may eventually develop into colorectal cancer (CRC). It may be fatal if colon cancer is discovered at this stage. Polyposis can occur singly, in groups, or as a component of polyposis syndrome. Food choices, lifestyle, physical activity, smoking, a positive family history, and other factors all have an impact on colon polyps. People over 50, those who are obese, and those who smoke frequently are more likely to develop colon polyps.

To obtain the CRC, a colonoscopy is essential. An integral component of colonoscopy is the detection of colon polyps [1] [2]. The successful identification and removal of adenomatous polyps linked to colorectal cancer (CRC) has been the primary cause of the 51 percent decline in adult CRC mortality and 32 percent decrease in adult CRC incidence rates over the past 50 years [3]. The detection rate of polyps still varies, though, and is impacted by a number of variables, including the endoscopists' skill and the features of the polyps themselves. Sometimes the small size of the polyps, the endoscopist's inexperience, or poor vision cause the polyp to go undetected. The little polyps became larger and developed into colorectal cancer (CRC) as a result. The intricate organ systems of the colon and rectum make them difficult to navigate and treat, requiring specialized knowledge from individuals. In a similar vein, polyp removal can be exceedingly challenging due to persistent organ deformations that can make it challenging to identify the lesion boundaries, making a full resection challenging and requiring the knowledge of endoscopic specialists. Two advantages of systems with computer assistance are decreased operator subjectivity and increased adenoma detection rates (ADR). Therefore, computer-aided identification and segmentation approaches can assist in localizing polyps and guiding surgical therapies (such polypectomy) by visualizing the locations and borders of the polyps. We can create a model to identify colon polyps, their surface, and the characteristics of each type of polyp by using machine learning, especially deep learning. Deep learning can recognize common colon landmarks and differentiate between a variety of unknown colon disorders.

1.2 Colorectal Polyp Identification

Colorectal polyps are abnormal growths in the lining of the colon or rectum, with significant implications for colorectal cancer risk, as certain types can develop into malignancies over time. Effective identification of these polyps is essential, and various screening methods are employed, including colonoscopy, which allows direct visualization and removal of polyps, and flexible sigmoidoscopy, which examines only the lower colon. Additionally, non-invasive techniques like CT colonography provide a valuable alternative for patients. Though many polyps are asymptomatic, symptoms such as rectal bleeding, changes in bowel habits, and abdominal pain may indicate their presence. Risk factors, including age (with increased risk starting at 50), familial history, genetic predispositions, and lifestyle choices, underscore the importance of regular screening. Thus, comprehensive screening approaches are crucial for the early detection and prevention of colorectal cancer, emphasizing the need for awareness and proactive health measures in at-risk populations.

1.3 Application of Deep learning-based model in colon polyp detection

There is a lot of promise for using machine learning, particularly deep learning, to detect and describe colon polyps and to improve the early detection and treatment of colorectal cancer. Applications for identifying and characterizing colon polyps include the following:

- **Early diagnosis and categorization of colon cancer**: By using machine learning algorithms on live video feeds or colonoscopy pictures, accurate colon polyp identification and characterization can be accomplished. This makes early identification easier, allowing for timely therapy and intervention. This allows us to determine the types of polyps and their stages, which may aid a medical practitioner in administering additional therapy [4].
- **Colon polyp identification and classification**: Historically, visual inspection has been employed to detect colon polyps during procedures such as colonoscopies. Thanks to advancements in technology, machine learning algorithms—particularly deep learning models—are revolutionizing this process. These algorithms locate polyps in real-time video feeds or colon pictures by using sophisticated pattern recognition. A more comprehensive detection process is ensured by their exceptional capacity to detect minute features and irregularities that could be invisible to the human eye. These models describe and classify polyps in addition to identifying them. This means categorizing polyps based

on their attributes, such as size, shape, texture, and other relevant factors. Classification facilitates the differentiation of potentially malignant growths, precancerous adenomas, and benign polyps. This comprehensive categorization is necessary to determine the appropriate course of action, whether it be instant removal, continued surveillance, or further study [5].

- **Classification of questionable tissue damage at the colon**: A variety of illnesses, such as tumors, lesions, ulcers, and inflammation, can cause suspicious tissue damage in the colon. To identify and diagnose these abnormalities, advanced imaging techniques such as colonoscopy and endoscopy are crucial. However, interpretations of these visuals might be subjective and subject to human mistake. A ground-breaking technique for identifying and categorizing these tissue anomalies is the use of machine learning, especially deep learning models. By analyzing images or real-time video feeds, these models excel at spotting even the smallest deviations from normal tissue, which aids in the early detection of any issues [2].

1.4 Motivation

Our research is driven by the pressing need to treat the health issues associated with colon polyps, which have the potential to progress into potentially lethal colorectal cancer. Despite colonoscopy's success in reducing mortality rates, issues such polyps that go undetected due to their size and the endoscopist's skill still persist. The goal is to use deep learning to enhance colon polyp recognition, surface characterization, and classification in order to provide a more reliable and consistent method. Benefits like higher adenoma detection rates and reduced operator subjectivity are expected. The objective is to offer a trustworthy computer-assisted technique for precise diagnosis of colorectal polyps.

1.5 Objective

The objective of the thesis paper is

- To design PolypNet as a shallow Convolutional Neural Network (CNN) enhanced with a self-attention mechanism.
- To compare the performance of PolypNet against established models like VGG16, ResNet50, DenseNetv3, and MobileNetv3.

- To address the limitations of existing models, specifically overfitting and poor generalization on small datasets.
- To improve the classification of colorectal polyps.

1.6 Challenges

The following are the difficulties in creating a deep learning-based model for the identification, surface description, and categorization of colorectal polyps:

- **Limited Datasets**: Training and testing machine learning models is difficult since there aren't many large, publicly accessible datasets that include a variety of patient demographics and imaging modalities. Models may not be altered as successfully if datasets do not adhere to the most recent standards [6].
- **Variability in Polyp Properties**: The size, shape, and appearance of colon polyps can vary greatly from one another. It is difficult to develop a model that can reliably identify and categorize this variability; a representative and varied dataset is needed [7].
- **Endoscopist expertise**: The system's effectiveness depends on how well endoscopists record and take clear pictures during colonoscopies. The effectiveness of the system may be impacted by differences in endoscopists' abilities and methods [8].
- **Privacy and Ethical Issues**: When handling medical data, particularly pictures and videos, patients' privacy and consent present ethical issues. It is crucial to design a system that complies with privacy regulations and preserves data integrity.
- **Occlusion**: When some body parts were hidden in photos, it was difficult to identify important details [9].

1.7 Organization of the thesis

In the next chapters, the many aspects of this thesis will be covered in greater detail. All of the issues will be discussed in the following sections of this report. The history of deep learning is covered in chapter 2. In chapter 3, we discussed the latest findings on this topic. The methods are presented in Chapter 4. In the next chapter 5, we discussed the results, the comparative analysis of the models, and the implementation of the proposed method. This thesis has been finished, and chapter 6 contains any further demands.

1.8 Summary

Colorectal polyps are mucosal protrusions in the colon, primarily benign but capable of developing into colorectal cancer (CRC). Early detection and removal of adenomatous polyps are critical for reducing CRC mortality rates, with risk factors including age (over 50), obesity, smoking, and family history. Colonoscopy is the main detection method, but its effectiveness can vary due to endoscopist skill and polyp characteristics, leading to missed diagnoses.

To address these issues, machine learning and deep learning technologies offer promising solutions for improving the identification and classification of colon polyps. This research aims to develop PolypNet, a shallow Convolutional Neural Network (CNN) model enhanced with a self-attention mechanism, to enhance classification accuracy and tackle limitations of existing models. Key challenges include the limited availability of diverse datasets, variability in polyp characteristics, and ethical concerns regarding patient privacy. By overcoming these obstacles, the study seeks to contribute to better detection and management of colorectal polyps, ultimately improving CRC prevention and treatment outcomes.

2. CHAPTER 2: THEORETICAL BACKGROUND

2.1 Overview

A collection of cells called colon polyps forms on the colon's epithelium. Colonic polyps can be discovered and removed during a colonoscopy before they develop into cancer. However, around one-fourth of the polyps can be missed due to their small size, location, or human mistake [10]. use a deep learning model capable of identifying and categorizing the polyp. One type of deep learning methodology is convolutional neural networks. The theoretical underpinnings of colon polyp identification and categorization were discussed in this chapter.

2.2 Deep Learning

One of the most common AI techniques used for processing big data is machine learning, a self-adaptive algorithm that gets increasingly better analysis and patterns with experience or with newly added data.

Traditional machine learning was confined is the way it processes data, as some functionalities needed some exactly specific programming to perform some specific tasks. The traditional machine-learning could not receive raw data as input and transform it into a suitable understandable representation without the help of human brains. A machine-learning algorithm required labeled/structured data to understand the differences between images of cats and dogs, learn the classification and then produce output.

On the contrary, a subset of machine learning where algorithms are created and function similar to those in machine learning, but there are numerous layers of these algorithms- each providing a different interpretation to the data it feeds on. It did not require any labeled/structured data, as it relied on the different outputs processed by each layer.

Deep learning is capable of using raw data and can automatically learn the features required to perform the specific identification task. The learning method can be supervised, semi-supervised, or unsupervised. This learning ability is based on stacking several non-linear modules as a stack of multiple layers that convert the raw input data into a higher label more abstract representation [5]. Each successive layer uses the output from the previous layer as input for the next layer. So, it is like a cascade of multiple layers.

It requires high-end machines contrary to traditional Machine Learning algorithms. GPU has become an integral part now to execute any Deep Learning algorithm. In traditional Machine learning techniques, most of the applied features need to be identified by a domain expert in order to reduce the complexity of the data and make patterns more visible to learning algorithms to work. The biggest advantage of Deep Learning algorithms as discussed before are that they try to learn high-level features from data in an incremental manner. This eliminates the need for domain expertise and hardcore feature extraction.

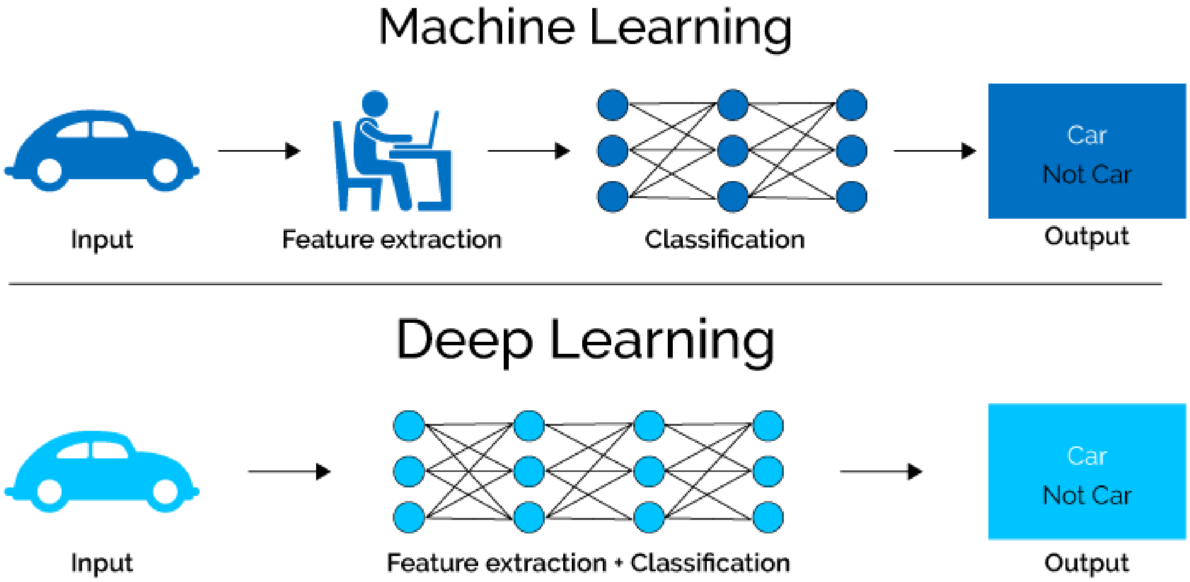


Fig. 2.1 Machine Learning vs. Deep Learning

2.3 Neural Network

A neural network, more properly referred to as an 'artificial' neural network (ANN), is provided by the inventor of one of the first neurocomputers, Dr. Robert Hecht-Nielsen. He defines a neural network as: “a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs.” A diagram of what one node might look like as shown in the graphic below.

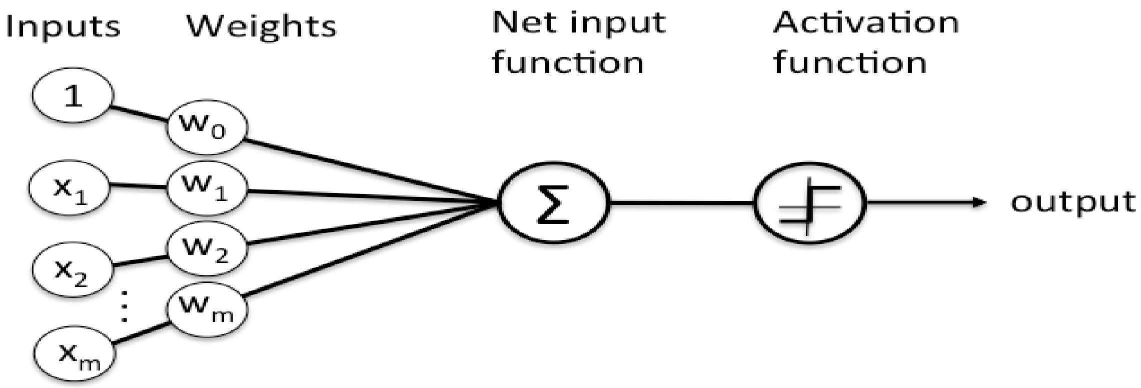


Fig. 2.2 Simple Neural Network

Neural networks are typically organized in layers. Layers are made up of a number of interconnected 'nodes' which contain an 'activation function'. Patterns are presented to the network via the 'input layer', which communicates to one or more 'hidden layers' where the actual processing is done via a system of weighted 'connections'. The hidden layers then link to an 'output layer' where the answer is output as shown in the graphic below.

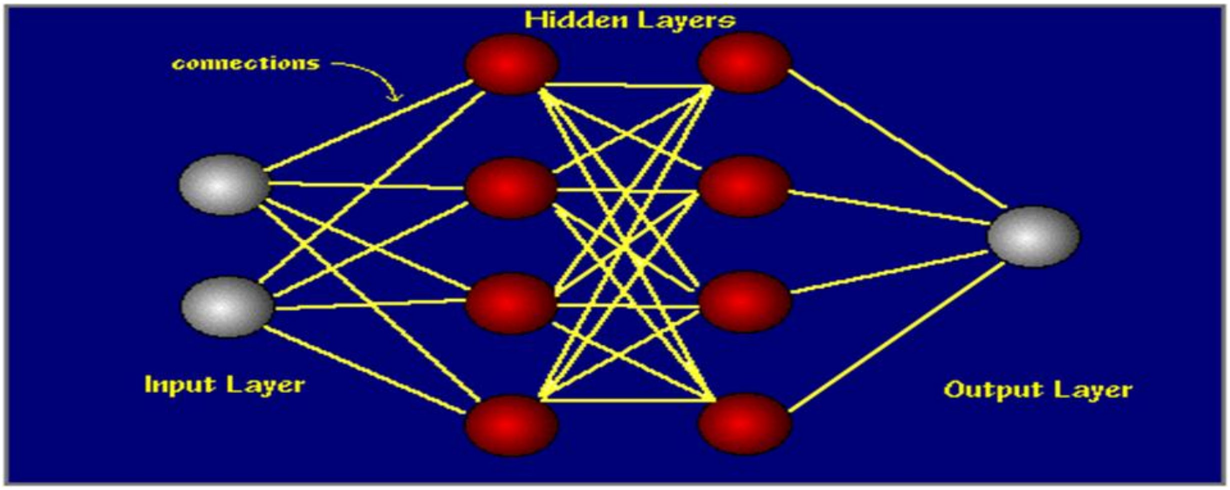


Fig. 2.3 Deep Neural Network with two hidden layers

A neural network is a set of connected neurons designed in layers:

- **a) Input Layer**: Brings the initial data into the system for further processing by subsequent layers of artificial neurons.
- **b) Hidden layer**: A hidden layer is a layer between input layers and output layers. In the hidden layer, artificial neurons take in a set of weighted inputs and give an output through an activation function.
- **c) Output layer**: In the program, the last layer of neurons that produces given outputs.

Neural networks may accomplish tasks that take hours or even days, such speech recognition and picture identification, in a matter of minutes. A popular illustration of a neural network is the search engine Google [11].

2.4 Convolutional Neural Network

Convolutional Neural Networks are very similar to ordinary Neural Networks from the previous section: they are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. The whole network still expresses a single differentiable score function: from the raw image pixels on one end to class scores at the other. And they still have a loss function (e.g., SVM/Softmax) on the last (fully-connected) layer and all the tips/tricks we developed for learning regular Neural Networks still apply.

A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels).

A simple ConvNet is a sequence of layers, and every layer of a ConvNet transforms one volume of activations to another through a differentiable function. Here three main types of layers are used to build ConvNet architectures: Convolutional Layer, Pooling Layer, and Fully-Connected Layer (exactly as seen in regular Neural Networks).

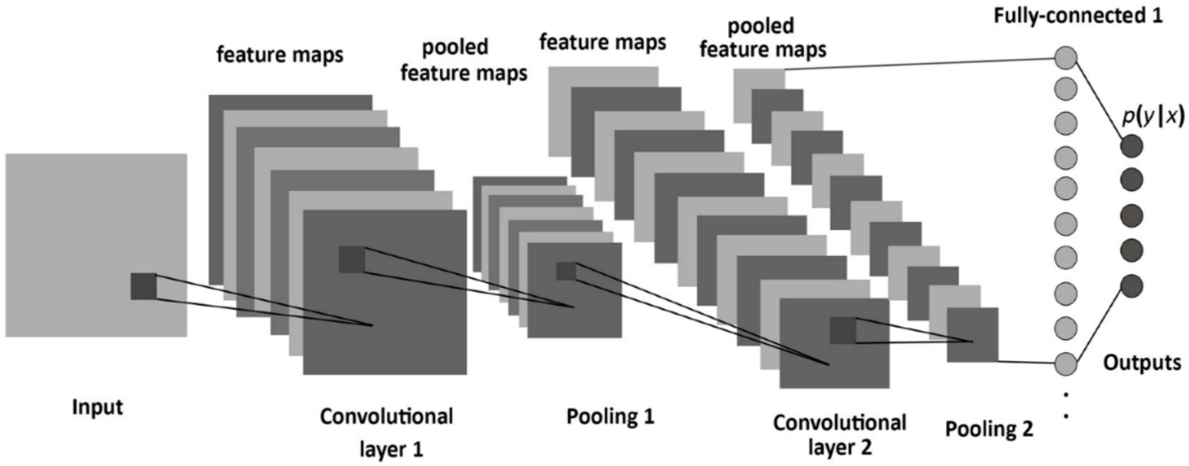
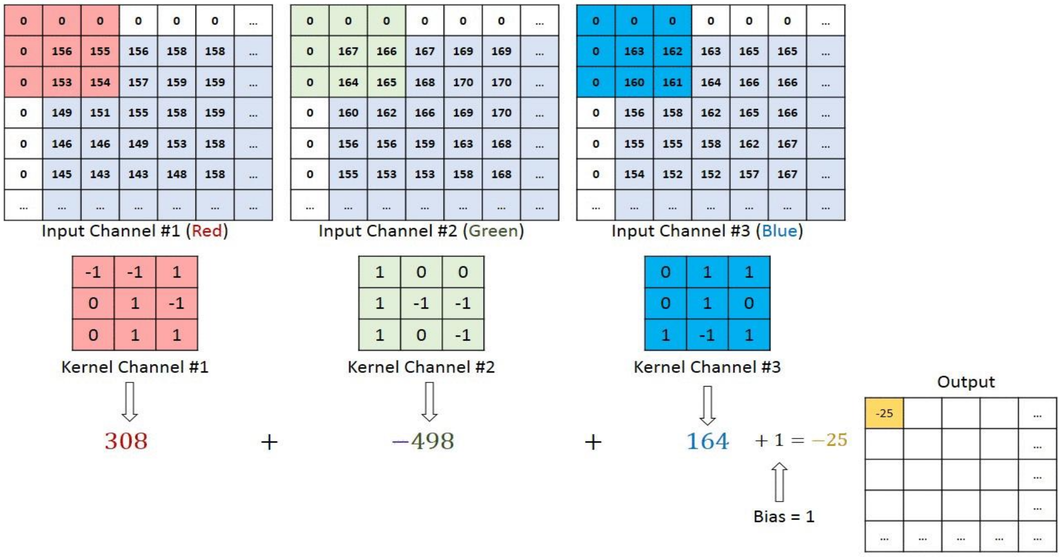


Fig. 2.4 Convolutional Neural Network

CNNs are used in autonomous number plate reading, photo recognition, and self-driving car software. Convolutional neural networks have gained a lot of popularity because of their adaptability to changes in data size, rotation, distortion, and other factors [12].

2.4.1 Convolutional Layer

The Conv layer is the core building block of a Convolutional Network that does most of the computational heavy lifting. Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel.



Input Channel #1 (Red)

0	0	0	0	0	0	...
0	156	155	156	158	158	...
0	153	154	157	159	159	...
0	149	151	155	158	159	...
0	146	146	149	153	158	...
0	145	143	143	148	158	...
...	...	...	...	...	...	...

Input Channel #2 (Green)

0	0	0	0	0	0	...
0	167	166	167	169	169	...
0	164	165	168	170	170	...
0	160	162	166	169	170	...
0	156	156	159	163	168	...
0	155	153	153	158	168	...
...	...	...	...	...	...	...

Input Channel #3 (Blue)

0	0	0	0	0	0	...
0	163	162	163	165	165	...
0	160	161	164	166	166	...
0	156	158	162	165	166	...
0	155	155	158	162	167	...
0	154	152	152	157	167	...
...	...	...	...	...	...	...

Kernel Channel #1

-1	-1	1
0	1	-1
0	1	1

Kernel Channel #2

1	0	0
1	-1	-1
1	0	-1

Kernel Channel #3

0	1	1
0	1	0
1	-1	1

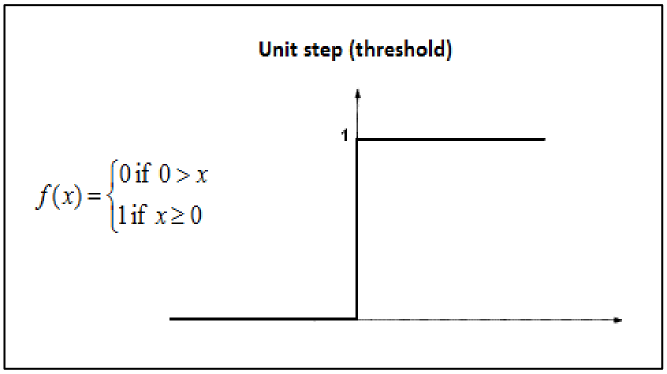
Output

-25				...
				...
				...
				...
...	...	...	...	...

Fig. 2.5 Convolution operation on an MxNx3 image matrix with a 3x3x3 Kernel

2.4.2 Activation Functions

- **Step function**: A step function is defined as



$$f(x)=\begin{cases}0\,\text{if }0>x\\1\,\text{if }x\geq 0\end{cases}$$

Fig. 2.6 Step function

Where the output is 1 if the value of x is greater than equal to zero and 0 if the value of x is less than zero. As one can see a step function is non-differentiable at zero. Since the step function is non-differentiable at zero hence it is not able to make progress with the gradient descent approach and fails in the task of updating the weights.

To overcome, this problem sigmoid functions were introduced instead of the step function.

- **Non-Linearity (ReLU)**: ReLU stands for the Rectified Linear Unit. This is the equation for ReLU:

$$ReLU(y) = max(0, x) \tag{2.1}$$

$$ReLU(y) = \begin{cases} 0 & if\ x < 0 \\ 1 & if\ x \geq 0 \end{cases} \tag{2.2}$$

The ReLU equation tells us this: If the input z is less than 0, set input equal to 0 and if the input is greater than 0, set input equal to the input.

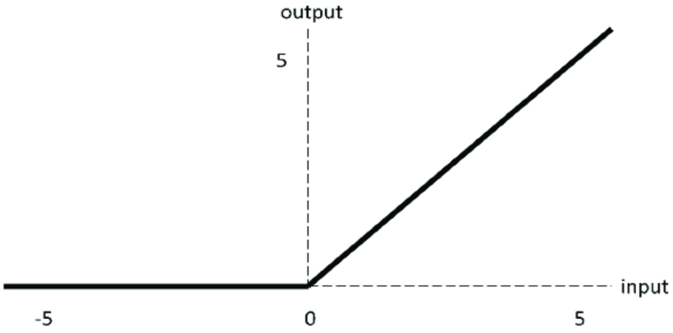


Fig. 2.7 ReLU activation function

- **Softmax**: Softmax is a very interesting activation function because it not only maps our output to a [0,1] range but also maps each output in such a way that the total sum is 1. The output of Softmax is, therefore, a probability distribution. The softmax function is often used in the final layer of a neural network-based classifier. Such networks are commonly trained under a log loss (or cross-entropy) regime, giving a non-linear variant of multinomial logistic regression.

2.4.3 Batch Normalization

Batch normalization is a technique for training very deep neural networks that standardizes the inputs to a layer for each mini-batch. This has the effect of stabilizing the learning process and dramatically reducing the number of training epochs required to train deep networks. During training time, a batch normalization layer does the following:

1) Calculate the mean and variance of the layer's input

$$\text{Batch mean, } \mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{2.3}$$

$$\text{Batch variance, } \sigma^2{}_B = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \tag{2.4}$$

Batch statistics for step 1

2) Normalize the layer inputs using the previously calculated batch statistics.

$$\bar{x} = \frac{x_i - \mu_B}{\sqrt{\sigma^2{}_B + \in}} \tag{2.5}$$

Normalization of the layer's input in step 2

3) Scale and shift in order to obtain the output of the layer.

$$y_i = \sqrt{x_i} + \beta \tag{2.6}$$

Scaling and shifting the normalized input for step 3

2.4.4 Pooling Layer

It is common to periodically insert a Pooling layer in-between successive Conv layer in a ConvNet architecture. Its function is to progressively reduce the spatial size of the representation to reduce the number of parameters and computation in the network, and hence to also control overfitting. The Pooling Layer operates independently on every depth slice of the input and resizes it spatially, using the MAX operation. The most common form is a pooling layer with filters of size 2x2 applied with a stride of 2 down samples every depth slice in the input by 2 along both width and height, discarding 75% of the activations. Every MAX operation would, in this case, be taking a max over 4 numbers (little 2x2 region in some depth slice). The depth dimension remains unchanged.

- **Max-Pooling**: Max pooling is used to reduce the image size by mapping the size of a given window into a single result by taking the maximum value of the elements in the window.

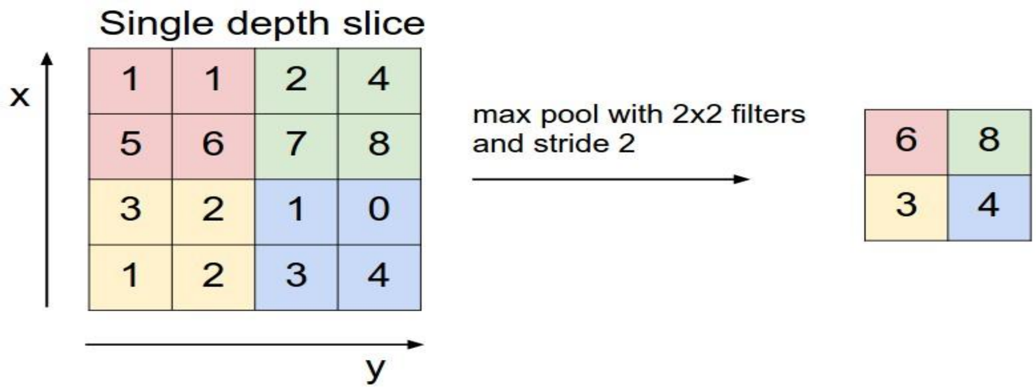


Fig. 2.8 Max-Pooling

- **Average-Pooling**: It's the same as max-pooling except that it averages the windows instead of picking the maximum value.

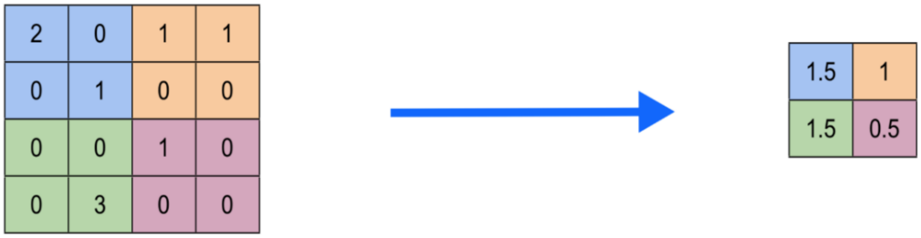


Fig. 2.9 Average-Pooling

2.4.5 Dropout

Dropout is a technique used to prevent a model from overfitting. Dropout works by randomly setting the outgoing edges of hidden units. A fully connected layer occupies most of the parameters, and hence, neurons develop co-dependency amongst each other during training which curbs the individual power of each neuron leading to over-fitting of training data.

2.4.6 Flatten Layer

Flatten is the function that converts the pooled feature map to a single column that is passed to the fully connected layer. Dense adds the fully connected layer to the neural network. Once the pooled featured map is obtained, the next step is to flatten it. Flattening involves transforming the entire pooled feature map matrix into a single column which is then fed to the neural network for processing.

2.4.7 Fully-Connected layer

Neurons in a fully connected layer have full connections to all activations in the previous layer, as seen in regular Neural Networks. Their activations can hence be computed with a matrix multiplication followed by a bias offset. Matrix is flattened into the vector and fed it into a fully connected layer like a neural network.

The fully connected (FC) layer in the CNN represents the feature vector for the input. This feature vector/tensor/layer holds information that is vital to the input. When the network gets trained, this feature vector is then further used for classification, regression, or input into other networks like RNN for translating into other types of output, etc. It is also being used as an encoded vector. During training, this feature vector is being used to determine the loss, and help the network to get train.

The convolution layers before the FC layer(s) hold information regarding local features in the input image such as edges, blobs, shapes, etc. Each conv layer holds several filters that represent one of the local features. The FC layer holds composite and aggregated information from all the conv layers that matters the most.

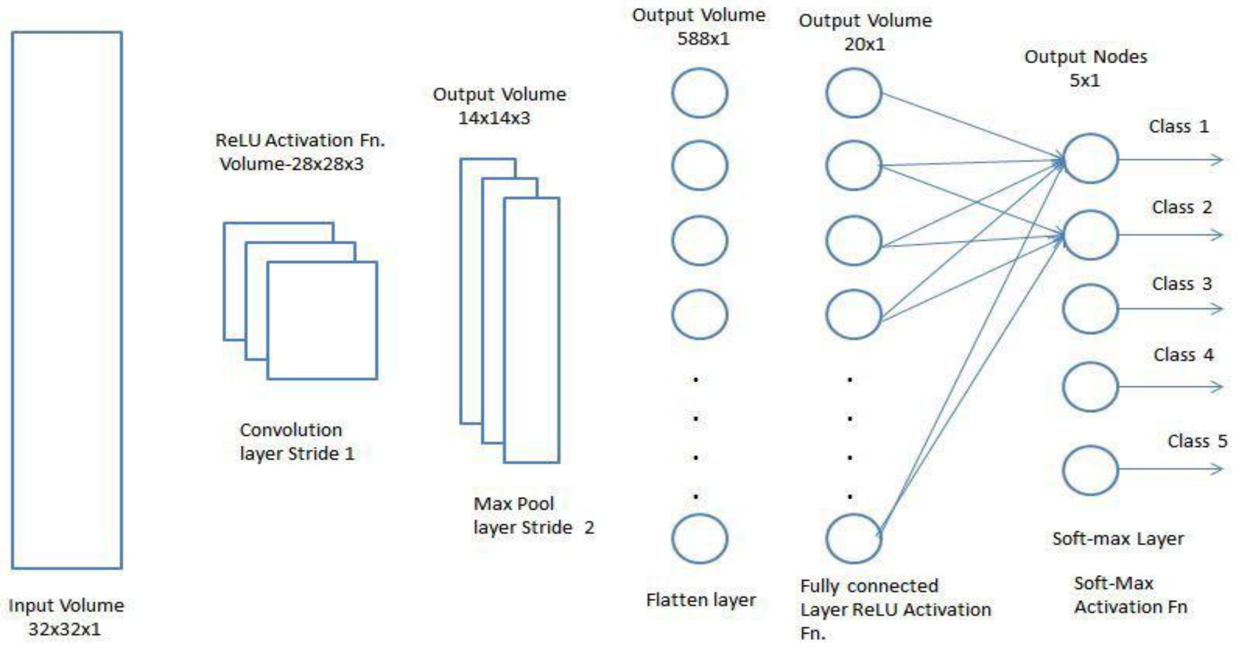


Fig. 2.10 Flatten layer and Fully Connected layer

2.5 VGG16

ConvNets are a kind of ANN. A network of CNN is composed of many hidden layers, an output layer, and an input layer. CNN like the VGG16 model, are considered to be among the top computer vision models (CVM) currently in use. The model's creators evaluated the networks and employed a tiny (3 x 3) convolution filter architecture to increase the depth, demonstrating a significant advancement over the prior state-of-the- art configuration [13].

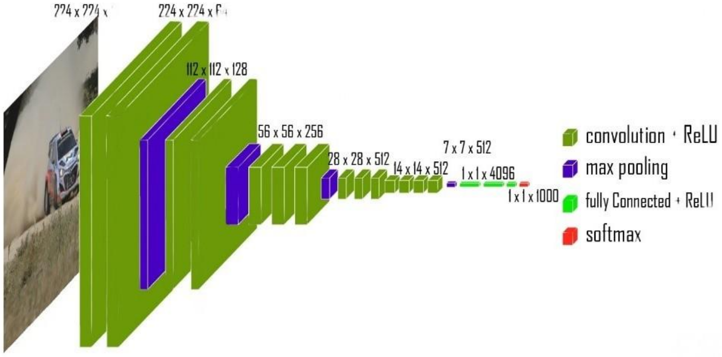


Fig. 2.11 VGG16 network architecture

2.5.1 VGG16 Architecture

- **Input**: The VGGNet model takes in an image of size 224×224. To ensure consistency during the ImageNet competition, it's cut 224×224 section from the middle of each image as the input.
- **Convolutional layers**: VGG makes use of 3×3 filters in its convolutional layers, which is the smallest possible size. Furthermore, the input undergoes 1×1 convolution filter.
- **ReLU activation**: One of the most significant advancements in reducing training time for AlexNet was the implementation of the Rectified Linear Unit Activation Function (ReLU). This innovative component behaves as a linear function, producing a zero value for negative inputs and an identical output for positive inputs. To preserve the spatial resolution during convolution, VGG employs a constant stride of 1 pixel.
- **Hidden Layers**: All of the hidden layers in VGG network utilize ReLU, as opposed to the implementation of Local Response Normalization in AlexNet. While the latter has minimal effect on overall accuracy, it significantly increases training time and memory consumption.
- **Pooling layers**: By including a pooling layer after a set of convolutional layers, the feature maps generated during each stage of convolution undergo a reduction in the number of

parameters and dimensionality. This is crucial, especially considering the rapid progression from 64 to 128, then to 256, and eventually to 512 filters in the concluding layers. Therefore, utilizing pooling is vital for optimizing the performance of the model.

- **Fully connected layers**: VGGNet consists of three interconnected layers, each with a distinct purpose. The first and second layers boast an impressive 4096 channels, while the third layer contains exactly 1000 channels, one for every class.

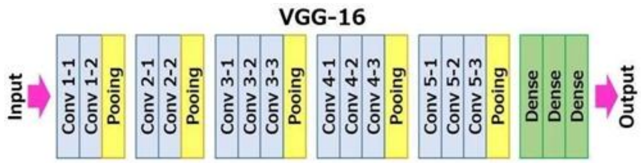


Fig. 2.12 VGG16 architecture Map

2.5.2 Object Localization in Image

We must replace the class score with the bounding box location coordinates during localization. A bounding box's location may be expressed as a 4-dimensional vector with the height, width, and center values corresponding to x and y. Both class-specific bounding boxes (which produce a 4-thousand-parameter vector) and shared bounding boxes (which produce a 4-parameter vector) are examples of the two forms of localization architecture. The article tested both strategies using the VGG16 (D) architecture. Additionally, in this case, we must switch from classification loss functions to regression loss functions, such as mean square error (MSE), which punish the difference between the predicted and actual losses.

2.5.3 Results

VGG16 performed among the top designs in the 2014 ILSVRC competition. It was only surpassed by GoogLeNet, which had a classification error of 6.66 %, finishing in the second spot, it boasted a remarkable 7.32 % top-5 classification error rate. It was also the task's victor, achieving a localization error of 25.32 %.

2.5.4 Limitation

- The original VGG model requires two to three weeks of training on an Nvidia Titan GPU.
- VGG16 prepared picture. The size of net weights is 528 MB. It is therefore inefficient because it uses a large quantity of storage space and bandwidth.
- An explosion of gradient problems arises with 138 million parameters.

2.6 ResNet50

Convolutional neural networks (CNNs) of the ResNet50 variety have completely changed the way we approach deep learning. Kaiming He et al. initially presented it at Microsoft Research Asia in 2015. ResNet, an acronym for residual network, describes the remaining building components that comprise the network's design. Deep residual learning is the foundation of ResNet50, which enables the training of extremely deep networks with hundreds of layers. A startling finding in deep learning research led to the creation of the ResNet architecture: a neural network's performance does not necessarily improve with the addition of more layers. This was surprising because a network should be able to learn more information in addition to what the prior network knew when a new layer is added. The ResNet group, under the direction of Kaiming He, created a unique design using skip connections to solve this problem. The network was able to learn more accurate representations of the input data thanks to these connections, which allowed knowledge from previous layers to be preserved. They were able to train networks with up to 152 layers using the ResNet design. With a 3.57 % mistake rate on the ImageNet dataset and victories in other contests, such as the ILSVRC and COCO object detection tasks, ResNet's results were revolutionary. This illustrated the ResNet architecture's strength and promise for use in deep learning studies and applications [14].

2.6.1 ResNet50 Architecture

ResNet50 is made up of 50 layers split up into 5 blocks, each of which has a collection of residual blocks. The network may learn more accurate representations of the input data by using the residual

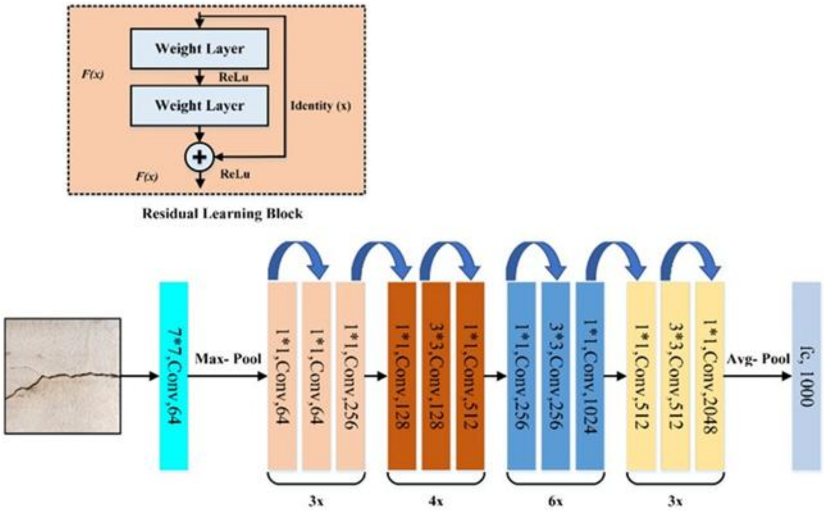


Fig. 2.13 ResNet50 Architecture

blocks, which enable the retention of information from previous levels [14]. The primary ResNET components are as follows.

- **Convolutional Layers**: Convolution on the input picture is carried out by the network’s first layer, the convolutional layer. A max-pooling layer that down samples the convolutional layer’s output comes next. Following the max-pooling layer, a number of residual blocks are applied to the output.
- **Residual Blocks**: The two convolutional layers that comprise a residual block are followed by a batch normalization layer and a rectified linear unit activation function. Next, the output of the second convolutional layer is mixed with the input of the residual block, which is then exposed to an additional ReLU activation function. The output of the residual block is then passed on to the next block.
- **Fully Connected Layer**: The fully connected layer, the final layer in the net- work, maps the output of the final residual block to the output classes. In the completely connected layer, the number of neurons and the number of output classes are equal.

2.6.2 Concept of Skip Connection

One important component of ResNet50 is skip connections, sometimes referred to as identity connections. They make it possible to keep data from previous layers, which aids in the network’s ability to learn more accurate representations of the input. By combining the output of an earlier layer with the output of a later layer, skip connections are created.

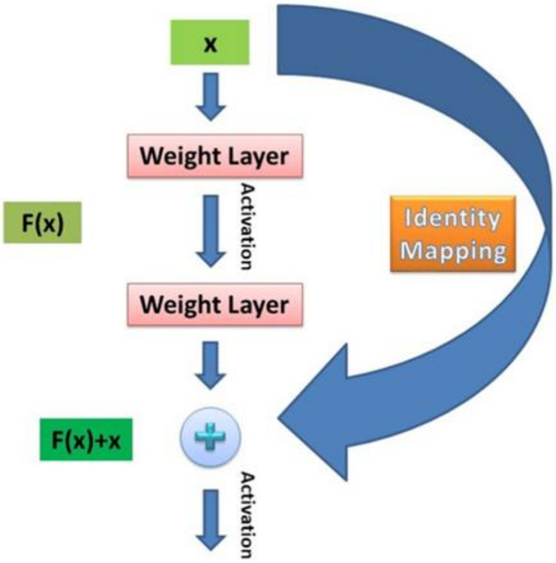


Fig. 2.14 Skip Connection

2.6.3 Advantages of ResNet50 Over Other Networks

ResNet50 has several advantages over other networks. One of its main advantages is its ability to train extraordinarily intricate networks with hundreds of layers. It is possible to retain data from earlier levels by using skip connections and remaining blocks. Another advantage of the model is ResNet50's ability to generate state-of-the-art results in a range of image-related tasks, such as object recognition, image classification, and picture segmentation.

2.7 DenseNetv3

DenseNet V3 is a convolutional neural network architecture that is based on the idea of densely connected layers. Each layer in a DenseNet V3 receives input from all the previous layers, which allows for better feature reuse and information flow. DenseNet V3 consists of several dense blocks, each containing a fixed number of convolutional layers, followed by transition layers that reduce the spatial dimensions and the number of feature maps [15]. DenseNet V3 is an improved version of DenseNet. DenseNet V3 uses a modified version of DenseNetv3, which is a 121-layer DenseNet with four dense blocks and three transition layers. DenseNet V3 also incorporates some techniques from YOLO- V3, a state-of-the-art object detection model, to enhance its performance on multi-scale remote sensing target detection. Some of these techniques include:

- Using a larger input size of 608 x 608 pixels, instead of the original 224 x 224 pixels, to capture more details of the targets.
- Adding a fourth detection scale at the end of the network, which outputs bounding boxes at a finer resolution of 76 x 76, instead of the original 19 x 19.
- Replacing the 1 x 1 convolution layer in the transition layer with a DenseNet layer, which increases the number of feature maps and the diversity of features.
- Applying a leaky ReLU activation function with a negative slope of 0.1, instead of a regular ReLU, to avoid gradient vanishing and improve the non-linearity of the network.

2.7.1 DenseNetv3 Architecture

In a DenseNet architecture, each layer is connected to every other layer, hence the name Densely Connected Convolutional Network. For L layers, there are L(L+1)/2 direct connections. For each layer, the feature maps of all the preceding layers are used as inputs, and its own feature maps are used as input for each subsequent layer.

This is really it, as simple as this may sound, DenseNets essentially connect every layer to every other layer. This is the main idea that is extremely powerful. The input of a layer inside DenseNet is the concatenation of feature maps from previous layers.

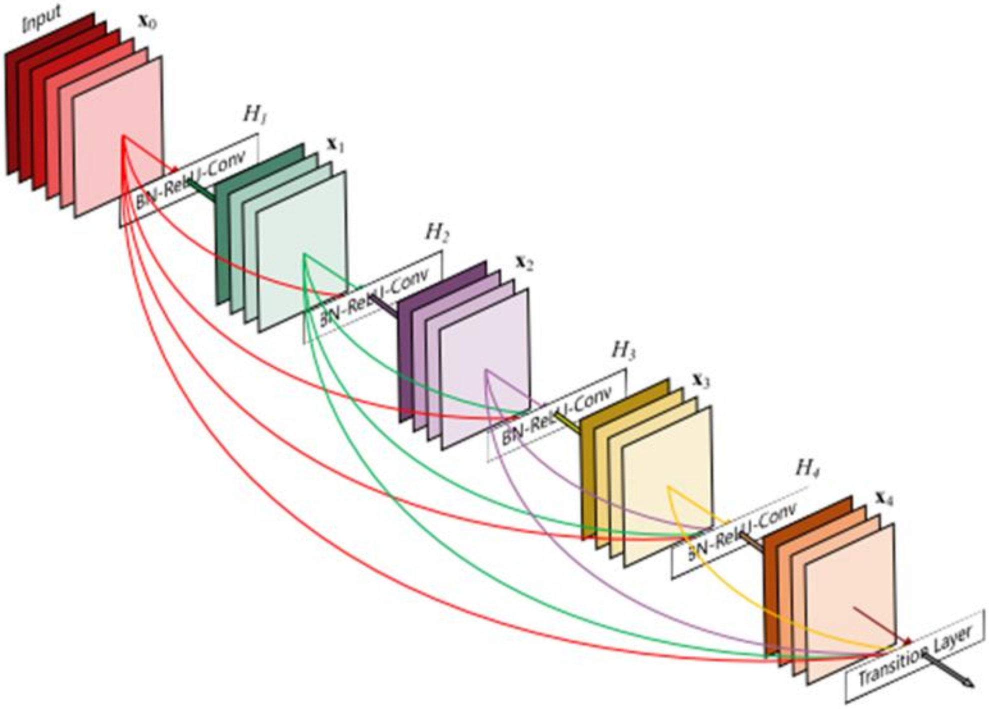


Fig. 2.15 DenseNetv3 Architecture

2.7.2 Advantages of DenseNetv3

Here are the advantages of DenseNetv3:

- **Improved Gradient Flow**: Alleviates the vanishing gradient problem.
- **Efficient Parameter Use**: Requires fewer parameters, enhancing memory efficiency.
- **Feature Reuse**: Allows richer feature representation by connecting all layers.
- **Strong Performance**: Achieves high accuracy on benchmark datasets.
- **Reduced Overfitting**: Minimizes overfitting risk, especially with small datasets.
- **Versatility**: Adaptable for various tasks, including segmentation and detection.
- **Modular Design**: Facilitates easy modification of network depth.
- **Context Awareness**: Captures multi-scale information for better object recognition.
- **Faster Inference**: Generally faster during inference compared to similar architectures.

2.7.3 Disadvantages of DenseNetv3

Here are the disadvantages of DenseNetv3:

- **Increased Computational Complexity**: Dense connections result in higher memory and processing power requirements.
- **Longer Training Times**: More complex architecture leads to extended training durations compared to simpler models.
- **Difficulties in Implementation**: The unique connectivity patterns can complicate implementation and tuning.
- **Sensitivity to Hyperparameters**: Requires careful adjustment of hyperparameters, which may necessitate extensive experimentation.
- **High Memory Consumption**: Storing multiple feature maps from all layers can lead to excessive memory usage during training.
- **Stage Bottlenecks**: Potential bottlenecks at certain stages due to processing large numbers of features, affecting efficiency.
- **Limited Interpretability**: The model's decisions may be difficult to interpret, making it hard to understand feature importance.
- **Complexity in Transfer Learning**: May complicate fine-tuning when adapting the model for new tasks or datasets.

2.8 MobileNetv3

MobileNetv3 is an advanced deep learning architecture designed specifically for mobile and edge devices. It builds upon the successful foundations of its predecessors (MobileNet V1 and V2), introducing optimizations that improve performance and efficiency while maintaining low latency and high accuracy [16]. Leveraging techniques such as neural architecture search (NAS), lightweight operations, and improved activation functions, MobileNetv3 is well-suited for real time applications like image classification, object detection, and semantic segmentation on resource-constrained devices.

2.8.1 MobileNetv3 Architecture

MobileNetv3 incorporates several key design choices and features, including:

- **Depthwise Separable Convolutions**: MobileNetv3 continues to use depth wise separable convolutions, which split the convolution operation into two smaller operations: a depth wise convolution (applying a single filter to each input channel) followed by a pointwise convolution (1x1 convolution to combine the outputs).

- **Neural Architecture Search (NAS)**: The architecture of MobileNetv3 was optimized using NAS, leading to an intelligent design that balances performance and computational efficiency.
- **Inverted Residual Blocks**: Similar to MobileNet V2, MobileNetv3 employs inverted residual blocks, which consist of a lightweight linear bottleneck structure that enhances feature extraction.
- **Activation Functions**: MobileNetv3 introduces the Swish activation function, which provides better performance compared to ReLU. Additionally, it utilizes Hard-Swish for faster computation while retaining benefits of Swish.
- **Squeeze-and-Excitation Blocks**: These blocks help to recalibrate channel-wise feature responses, improving the model's representational power.
- **Multi-Branch Architecture**: MobileNetv3 employs a multi-branch design in its architecture, allowing it to capture more diverse features while keeping the overall model lightweight.
- **Version Variants**: MobileNetv3 comes in two variants:
    - **MobileNetv3-Large**: Optimized for higher accuracy but requires more computational resources.
    - **MobileNetv3-Small**: A more lightweight version, optimized for faster performance and efficiency.

2.8.2 Advantages of MobileNetv3

- **High Efficiency**: MobileNetv3 achieves a strong balance between accuracy and computational efficiency, making it ideal for mobile devices.
- **Reduced Model Size**: The architecture is designed to minimize the number of parameters, facilitating faster downloads and lower memory usage.
- **Real-Time Performance**: Optimized for lower latency, enabling real-time processing for applications like image classification and object detection on edge devices.
- **Versatile Applications**: Suitable for various tasks, including image recognition, object detection, and segmentation, due to its efficient design.

- **Flexible Design**: The ability to choose between MobileNetv3-Large and MobileNetv3-Small allows developers to tailor the model based on specific application requirements.

2.8.3 Disadvantages of MobileNetv3

- **Limited Performance on Complex Tasks**: While efficient, MobileNetv3 might not perform as well as larger architectures (e.g., ResNet, DenseNet) in handling highly complex tasks that require deeper networks.
- **Sensitivity to Data Quality**: The model's performance may degrade significantly with poor quality or imbalanced datasets, necessitating careful data handling.
- **Interpretability Issues**: Like many deep learning models, MobileNetv3 can be seen as a "black box," making it difficult to interpret the reasoning behind its predictions.
- **Hardware Limitations**: Although designed for mobile devices, performance may vary depending on specific hardware capabilities, impacting the feasibility of some applications.

2.9 Summary

Finally, an examination of VGG16, ResNet50, MobileNetV3, and DenseNetv3 demonstrates the distinctive qualities of each network architecture and the advances they enable in computer vision. Our selection of models for certain tasks is influenced by our understanding of deep layer topologies, skip connections, inception modules, depth-wise separable convolutions, and dense connectivity. Our empirical inquiry is guided by this theoretical framework, which also helps with model selection and result interpretation. Recognizing the advantages of each architecture places our study in the context of deep learning’s ongoing evolution and highlights its contributions to the advancement of computer vision and image processing.

3. CHAPTER 3: LITERATURE REVIEW

3.1 Overview

Colon Polyp detection is an important technique to find out those polyps and characterized them before they become cancerous. Inclusion of deep learning, CNN and AI in this field it significantly increased the efficiency and accuracy of detecting polyps. Many researchers work on this to make it more accurate and efficient by using various kinds of methods based on Deep Learning and CNN like GAN, ViT, AlexNet, GoogLeNet, ResNet50, VGG16, VGG19, sECANet CAD and ResUNet++ architecture. Literature review of Colon Polyp detection using Deep learning and CNN are presented in this chapter.

3.2 Literature Review Based on Deep Learning

We studied and analyzed on this Colon Polyp detection and implementation of Deep learning and CNN on this novel technique. It is a novel field to make contributions.

3.2.1 Study 1

Computer-aided automated diminutive colonic polyp detection in colonoscopy by using deep machine learning system; first indigenous algorithm developed in India

In [10] developed a native artificial intelligence (AI) system compatible with any endoscopic video-capture software and high-definition colonoscopy for the purpose of identifying tiny polyps in real-world scenarios. To find and detect colonic polyps, a convolutional neural network model based on masked regions was constructed.

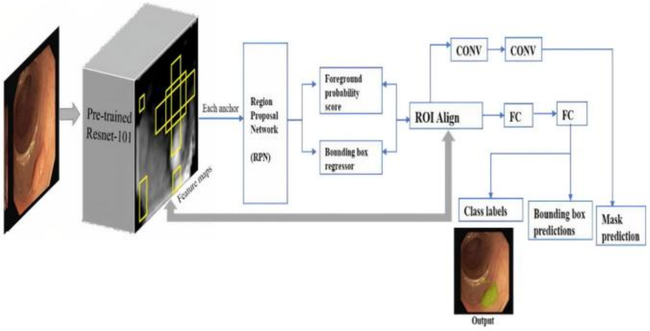


Fig. 3.1 Schematic of mask R-CNN network trained for polyp detection.

Three distinct datasets of colonoscopy recordings totaling 1,039 picture frames were used in total; these were divided into two groups: a training dataset consisting of 688 frames and a testing dataset consisting of 351. A total of 1,039 picture frames, 231 of which were derived from colonoscopy videos at our center. The remaining image frames were taken from previously modified, publicly available image frames that could be used immediately to construct the AI system. 88.63 %was the mean average precision (equivalent to specificity) for the AI system for autonomous polyp diagnosis. AI was able to detect each and every polyp in the tests because there were no false-negative results in the testing dataset (sensitivity of 100 %). The average size of the polyps in the study was 5 (±4) mm. Each photo frame took an average of 96.4 minutes to process.

Table 3.1 Summary of the study result

Dataset	True positives	False positives	False negatives	Precision (%) Testing dataset	Recall (%)	mAP (%)/ Specificity (%)
Original Dataset	299	315	46	48.69	86.6	79.56
Augmented Dataset	318	217	27	59.43	92.17	88.63

It might be possible to improve the AI system by continuously giving it new datasets to train on. This algorithm cannot be used in real time during a colonoscopy; it can only be applied to videos that have already been recorded. More datasets and/or a more powerful graphics processor could be used to reduce processing time during training.

3.2.2 Study 2

A novel machine learning-based algorithm to identify and classify lesions and anatomical landmarks in colonoscopy images

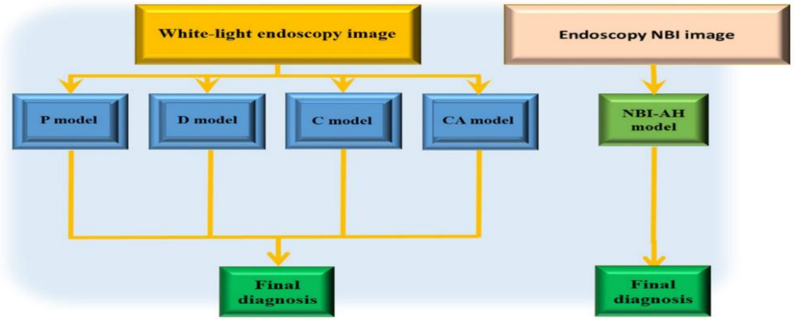


Fig. 3.2 AI-based GUTAID system.

In [17] identified and categorized anatomical landmarks and lesions in colonoscopy im- ages using a method based on machine learning that was demonstrated. To detect various colon lesions/landmarks, they developed a convolutional neural network (CNN)- based system called GUTAID, which had accuracy rates of 93.3 % for polyps, 93.9 % for diverticula, 91.7 % for cecum, 97.5 % for malignancy, and 83.5 % for adenomatous/hyperplastic polyps.

3.2.3 Study 3

Artificial intelligence for colonoscopy: Past, present, and future

In [18] discussed the gap between desired clinical attributes and currently available state-of-the-art technology in a session titled "Artificial intelligence for colonoscopy: Past, present, and future." It also offered potential directions for the development of endoscopic AI technologies that will bridge this divide. It suggested that in order to improve accuracy, the existing AI model be trained utilizing a substantial real-world dataset. Additionally, AI-assisted systems need to show that their implementation is less at CRC benchmarks.

3.2.4 Study 4

Automated colorectal polyp detection based on image enhancement and dual-path CNN architecture

In [7] proposed an innovative method to automatically detect intestinal polyps from colonoscopy images using the DP-CNN model. The method is applicable for real-time applications due to lower complexity and fewer learnable parameters than required by other existing methods. However, the authors suggest that the method can be made more accurate and effective with more datasets and different datasets from different regions and also on real time data.

3.2.5 Study 5

Real-time polyp detection model using convolutional neural networks

In [9] developed a deep learning model for real-time polyp detection based on a pre- trained YOLOv3 (You Only Look Once) architecture and complemented with a post- processing step based on an object-tracking algorithm to reduce false positives is re- ported, which could be integrated, in the future, into a CAD system. However, the authors suggest that the sensitivity could be improved by increasing the number of samples of the less frequent polyp histology and morphologies. Finally, in order to improve model validation, the model can be tested with public datasets, such as ETIS-Larib, and it can also be tested under a clinical trial.

3.2.6 Study 6

Automated Colorectal Polyp Classification Using Deep Neural Networks with Colonoscopy Images

In [19] presented two deep learning models, baseline and DeepCPC, that automate and facilitate the procedure of polyp classification with colonoscopy images.

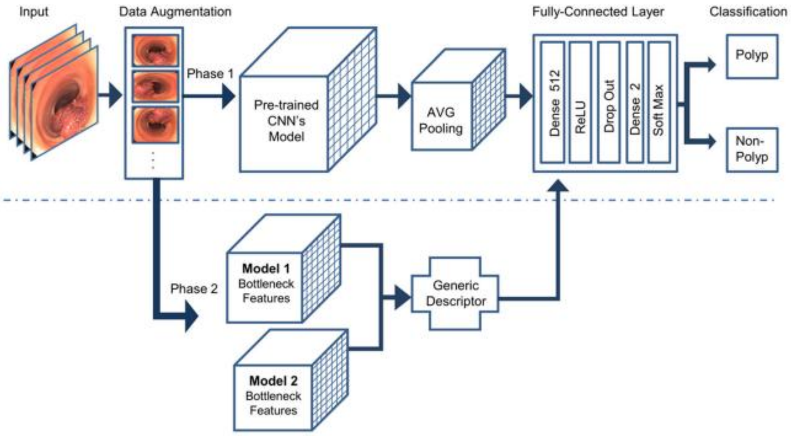


Fig. 3.3 A pictorial framework of the proposed DeepCPC model

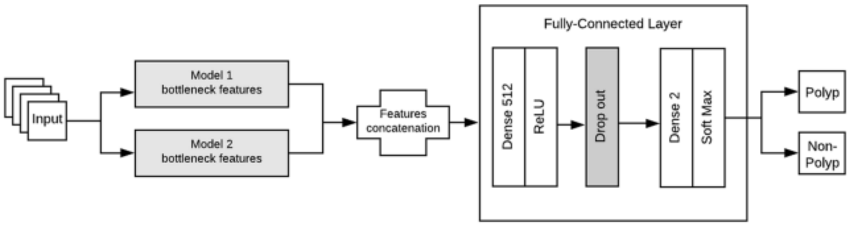


Fig. 3.4 The proposed procedure of features concatenation.

These models classify colorectal polyps based on the discriminative features extracted from the deep convolutional layers. The authors suggest that several improvements could be considered in the future, such as combining the dataset with different colorectal datasets to challenge the proposed model and including different polyps such as serrated sessile, pedunculated, tubular, to widen the area of use.

3.2.7 Study 7

Colonoscopy polyp detection and classification: Dataset creation and comparative evaluations

In [4] collaborated with experienced gastroenterologists to annotate the ground truth of polyp location and classification results, assembled an endoscopic dataset from many sources, and evaluated the performance of eight state-of-the-art deep learning-based object recognition models. According to the results, deep CNN models have potential for use in colorectal cancer screening. Our findings can serve as a baseline for future research on polyp identification and classification. The majority of the best-performing CNN models were developed before epoch 10. With 130k iterations, or roughly 45 epochs, RefineDet one-class detection is an anomaly. As contrast to 130k iter, 88.12 %, it has, nevertheless, reached comparable validation performance as early as 30k iter, 88.05 % mAP. One day, the dataset might be utilized for computer-aided diagnosis of colorectal cancer.

3.2.8 Study 8

Real-time detection of colon polyps during colonoscopy using deep learning: systematic validation with four independent datasets

In [20] developed a deep-learning algorithm to confirm the feasibility of an artificial intelligence system for automatic polyp detection during colonoscopy. They tested the performance of the algorithm using unaltered colonoscopy videos after systematic validation using two datasets of still images and one independent video dataset. The authors suggest that further clinical validation studies with large external video datasets are warranted to evaluate the generalizability of the algorithm in real-world colonoscopy practice.

3.2.9 Study 9

Colorectal Polyp Classification from White-light Colonoscopy Images via Domain Alignment

While many AI systems for colorectal polyp classification (CPC) depend on Narrow-Band Imaging (NBI) for high accuracy, their clinical applicability is limited as white-light (WL) colonoscopy remains the standard in many practice settings. Addressing this technological lag, Wang et al. [24] proposed an innovative solution using a teacher-student architecture to enable accurate classification directly from WL images. Their method trains a teacher network on NBI data and then uses domain alignment and contrastive learning to guide a student network to extract similarly rich features from WL images alone. This approach, validated on the first public paired WL-NBI dataset, achieved a significant 5.6% improvement in accuracy, demonstrating a viable path to deploying robust computer-aided diagnosis without relying on advanced imaging hardware.

3.2.10 Study 10

Semi-supervised Bladder Tissue Classification in Multi-Domain Endoscopic Images

The study by Lazo et al. [25] addresses a common clinical problem in computer-assisted diagnosis: the lack of annotated data across multiple imaging domains. Focusing on bladder tissue classification during Trans-Urethral Resection of Bladder Tumor (TURBT) procedures, the authors tackle the specific scenario where labeled data is available only for White Light Imaging (WLI), with no paired equivalent images in Narrow Band Imaging (NBI). To overcome this, they developed a semi-supervised GAN-based framework. This method uses a teacher network trained on labeled WLI data and a cycle-consistent GAN to perform unpaired image-to-image translation between WLI and NBI domains. A student network then leverages both real and synthetic images to learn robust feature representations. The model achieved high classification performance (accuracy: 0.90) on the labeled WLI domain and successfully generalized to the unlabeled NBI domain (accuracy: 0.92), demonstrating that reliable tissue classification is feasible even when annotations are limited to a single imaging modality. The authors also confirmed that the quality of the synthetically generated images was high enough to deceive medical specialists.

3.2.11 Study 11

Hierarchical Self-supervised Augmented Knowledge Distillation

Yang et al. [26] propose a novel knowledge distillation (KD) framework, HSAKD, designed to enhance the transfer of knowledge from a teacher to a student network. The authors identify two key limitations in previous methods: first, that some self-supervised tasks can interfere with the primary classification objective, and second, that most approaches only transfer knowledge from the final network layer, neglecting intermediate features. To address this, their method introduces a self-supervised augmented task that is learned jointly with the original classification task, creating a richer "joint distribution" knowledge that improves representations without harming classification performance. Furthermore, they implement a harchical distillation strategy by attaching auxiliary classifiers to intermediate layers, enabling diverse, one-to-one knowledge transfer throughout the network. This approach significantly outperforms previous state-of-the-art methods, demonstrating an average improvement of 2.56% on CIFAR-100 and 0.77% on ImageNet.

3.2.12 Study 12

Plant leaf disease classification using EfficientNet deep learning model

The study by Atila et al. [27] investigates the application of deep learning for automating the diagnosis of plant diseases from leaf images, a task traditionally reliant on slow and expert-dependent manual observation. To address the limitations of classical machine learning that require meticulous manual feature extraction, the authors propose the use of the EfficientNet architecture, which eliminates the need for such pre-processing. They evaluated EfficientNet models against other state-of-the-art deep learning architectures using the PlantVillage dataset, applying a transfer learning approach where all model layers were trainable. Their results demonstrated that the EfficientNet models, specifically the B5 and B4 variants, achieved superior performance, with the B4 model reaching a peak accuracy of 99.97% and a precision of 99.39% on an augmented dataset, significantly outperforming the other compared models.

3.2.13 Study 13

Multi-classification of breast cancer histopathological image using enhanced shallow convolutional neural network

Yusuf et al. [28] address the challenges of computational cost and training time associated with deep convolutional neural networks (DCNNs) for the multi-classification of breast cancer histopathological images. Noting that existing DCNN-based solutions are often hindered by high resource utilization and long convergence times, the authors propose an Enhanced Shallow Convolutional Neural Network (ES-CNN). This custom architecture was specifically designed to improve classification performance while minimizing computational demands. Evaluated on the BreakHis dataset across eight cancer types and four magnification factors, the ES-CNN model demonstrated both high accuracy and efficiency. The proposed method achieved multi-classification accuracies of 96%, 95%, 98%, and 96% at 40×, 100×, 200×, and 400× magnifications, respectively, successfully balancing state-of-the-art performance with reduced computational utilization.

3.2.14 Study 14

Dermatologist-level classification of skin cancer with deep neural networks

In their landmark study, Esteva et al. [29] demonstrated that a deep convolutional neural network (CNN) could achieve dermatologist-level performance in classifying skin cancer from clinical images. The researchers trained a single CNN end-to-end on a massively scaled dataset of 129,450 clinical images encompassing 2,032 different skin diseases. The model's performance was rigorously evaluated against 21 board-certified dermatologists on two critical binary classification tasks: distinguishing keratinocyte carcinomas from benign seborrheic keratoses, and malignant melanomas from benign nevi. The results showed that the AI system performed on par with all tested experts in both cases. This breakthrough established that deep learning could match the diagnostic accuracy of skilled dermatologists for specific, high-stakes classifications directly from images, highlighting the potential for AI-powered mobile devices to provide widespread, low-cost access to vital diagnostic care.

3.2.15 Study 15

MTANet: Multi-Task Attention Network for Automatic Medical Image Segmentation and Classification

Ling et al. [30] address the computational complexity of traditional two-stage clinical workflows, where segmentation and classification are performed separately, by proposing a one-stage Multi-Task Attention Network (MTANet). This unified model simultaneously performs high-quality medical image segmentation and disease classification. The architecture incorporates specialized attention modules: a Reverse Addition Attention module to fuse global and boundary features for precise segmentation, and an Attention Bottleneck module to integrate image features with clinical data for improved classification. When evaluated across three imaging modalities—polyp segmentation in colonoscopy (CVC-ClinicDB), skin lesion segmentation (ISIC-2018), and liver tumor segmentation/classification in ultrasound—MTANet outperformed state-of-the-art models. Notably, for liver tumor diagnosis, the proposed model demonstrated performance superior to that of all 25 radiologists in the study, highlighting its significant potential for efficient and accurate computer-aided diagnosis.

3.2.16 Study 16

Classification of Histopathological Biopsy Images Using Ensemble of Deep Learning Networks

Hosseinzadeh Kassani et al. [31] propose an ensemble deep learning approach to improve the automatic binary classification of breast cancer histology images. To enhance feature representation and classification accuracy, the authors combined three pre-trained Convolutional Neural Networks (CNNs)—VGG19, MobileNet, and DenseNet—into a single ensemble model. This model was used for feature extraction, after which the features were fed into a multi-layer perceptron for the final classification. The methodology incorporated several advanced techniques, including stain normalization, data augmentation, and hyperparameter fine-tuning. The proposed ensemble system was rigorously validated on four public benchmark datasets (BreakHis, ICIAR, PatchCamelyon, and Bioimaging), where it consistently demonstrated superior performance by achieving higher accuracy than individual single-model classifiers and traditional machine learning algorithms.

3.3 Summary

We reviewed different recent papers related to our work. Colon Polyp detection using deep learning and CNN is a novel field to work. It can help to change the medical field very much. We learned different types of Deep learning and CNN based approaches for our thesis. Here we analyzed and reviewed papers based on Deep learning and CNN.

4. CHAPTER 4: METHODOLOGY

4.1 Overview

For our thesis, we used a convolutional neural network to classify different types of polyps from image data. For training, we showed the data that we used for our model in 4.2 (dataset). The statement of our model is represented in 4.3 (our model). Our proposed model is described in 4.4 (our proposed model). Our result is shown in 4.5 (output).

4.2 Datasets

For our experiment, we have used kvasir-v1 dataset [21] for classification of different types of polyps from colonoscopy images. This is a la- belled dataset that contains 8 classes, and each class has 500 images at equal dimensions. The name of each class in the dataset is presented in Table 4.1. The number of images at each class is shown in Fig. 4.1 and sample images of different classes is shown in Fig. 4.2.

Table 4.1 Label information of our dataset

Label	**Class**
0	dyed-lifted-polyps
1	dyed-resection-margins
2	esophagitis
3	normal-cecum
4	normal-pylorus
5	normal-z-line
6	polyps
7	ulcerative-colitis

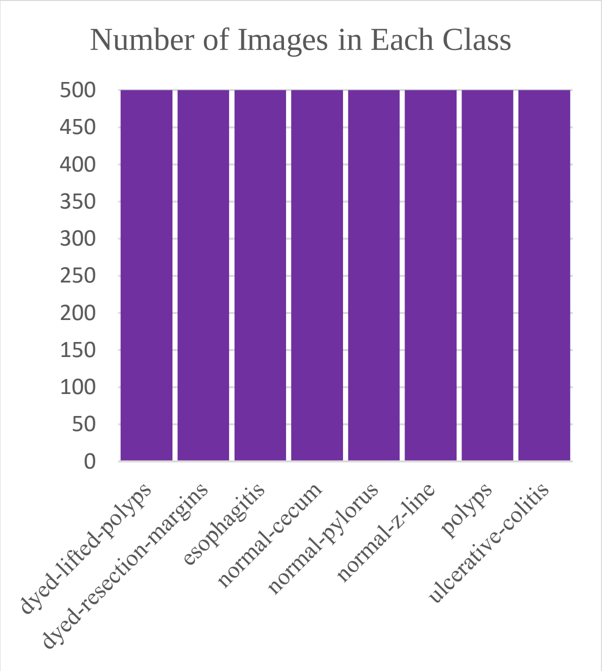
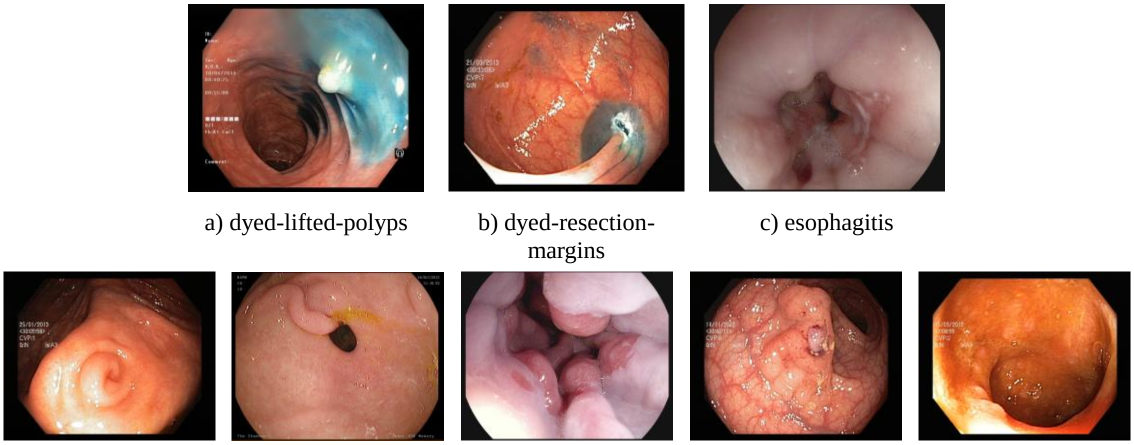


Fig. 4.1 Dataset image Details



a) dyed-lifted-polyps

b) dyed-resection-margins

c) esophagitis

d) normal-cecum

e) normal-pylorus

f) normal-z-line

g) polyps

h) ulcerative-colitis

Fig. 4.2 Sample images of different classes of the Kvasir-v1 dataset

4.3 Model Explanation

We used some state-of-the-art deep-learning CNN models to train and test our dataset. These are VGG16, ResNet50, DensNetV3, and MobileNetV3. We already describe those models in the theoretical background chapter. So, here we will discuss how we train our models.

4.3.1 Work-flow Diagram

We proposed a shallow CNN based method with self-attention mechanism named PolypNet, for reliably classifying colorectal polyp images and overcoming the overfitting problem. It considers the significant variations within the same class and the similarities across different classes. To assess the performance of the proposed model, it is compared to a few notable DL models. To train and assess the models, we need to divide the process into five steps: dataset preprocessing, data splitting, constructing model architecture, training the model, and evaluating the learned model. The proposed model is presented in Fig. 4.3.

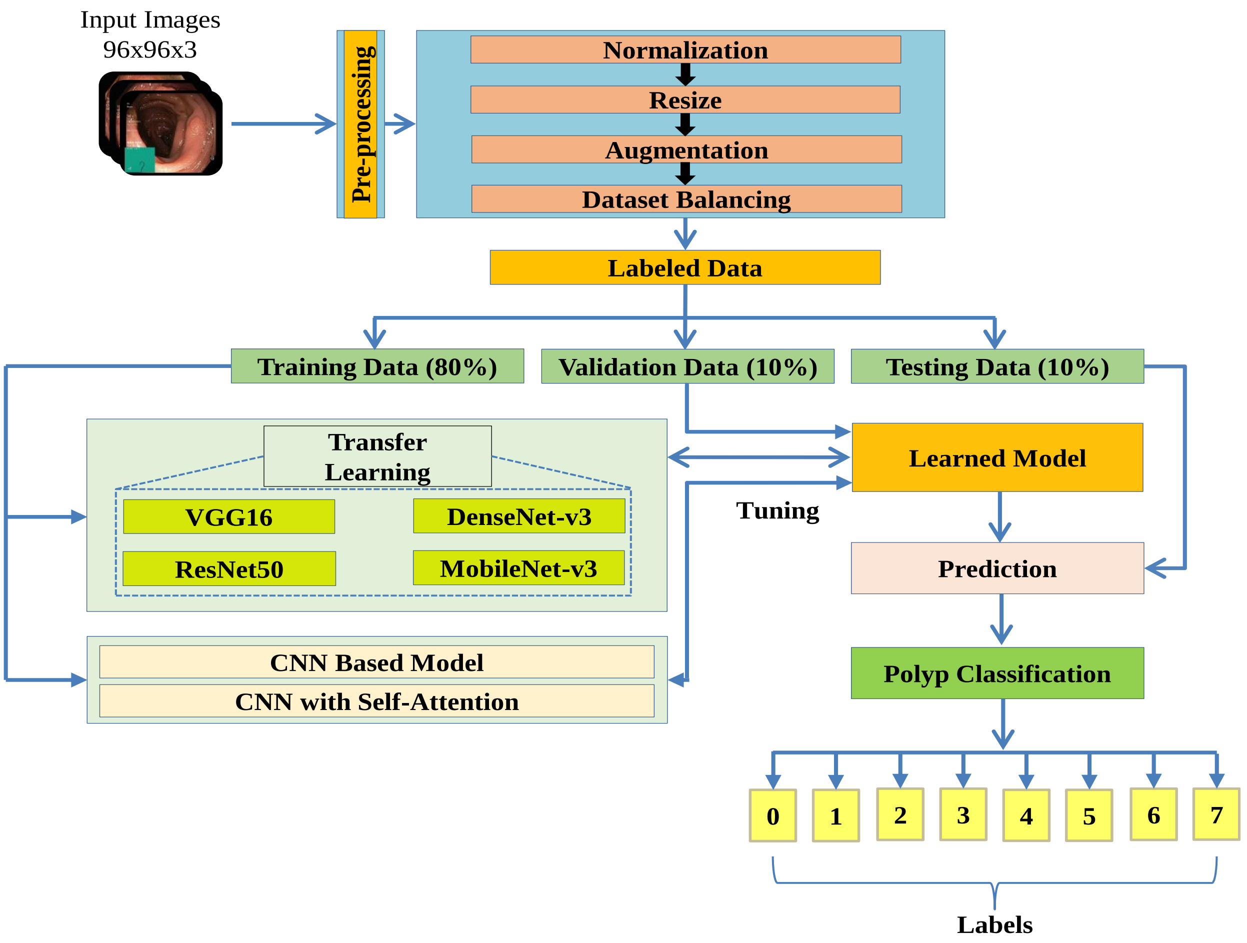


Fig. 4.3 The workflow diagram of the proposed method for colon polyp classification.

4.3.2 Prepossessing

Data preprocessing is crucial for maintaining the efficiency and reliability of deep learning models, especially in medical image segmentation. Data preprocessing begins with image normalization, which involves standardizing the pixel values across the entire dataset. The min-max normalization approach ensures consistent intensity distributions in medical images, reducing the impact of differences in sizes or scales between features. We utilize the augmentation technique to enhance the dataset, incorporating intentional activities such as resizing, zooming, rotation, shearing, flipping, and translation. The module mitigates overfitting and improves model resilience to image appearance changes. This study's images were resized at 96x96 pixels, ensuring computational resource optimization and maintaining uniformity during model training.

4.3.3 Data Splitting

To train and test the state-of-the-art CNN models in our study, data splitting is an essential step. We create three sets—one each for training, validating, and testing from the dataset. The largest section used to back-propagate and gradient descent for adjusting the parameters of a model is the training set. The set of validation data modifies the hyper-parameters to prevent over-fitting, enhancing the accuracy of the model. The set of tests evaluates how well the model performs on fresh data. The dataset splitting ratio is shown in the Table 4.2 below.

Table 4.2 Dataset Splitting

Training Set	Validation Set	Testing Set
80%	10%	10%

4.3.4 Transfer Learning Models

CNNs are known for their exceptional image classification capabilities because of their large parameter base and several hidden layers. These networks have a high degree of competence in identifying complex visual qualities and remain invariant to translation, in addition to their capaci-

Table 4.3 a short overview of a few transfer-learning models

Model	Architecture	Significance	Drawback	Variants used
VGG	Deep convolutional network	relatively straightforward, deeper networks with smaller filters	large number of parameters and significant computational resources required	VGG16
ResNet	Deep residual network	ability to mitigate vanishing gradient problems and learn more complex patterns.	prone to over fitting, require significant computational resources	ResNet50
DenseNet	Multi-layer CNN including dense blocks	reduce the number of parameters needed compared to traditional CNNs of similar depth	Complex architecture	DenseNetv3
MobileNet	Optimized CNN including dense blocks	ideal for mobile and embedded devices with limited computational power	generally, have lower accuracy compared to larger, more complex models	MobileNetv3

-ty to recognize spatial patterns in images. Leading CNN-based transfer learning architectures have received a lot of self-attention for their remarkable results in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), including VGG16[13], ResNet50 [14], DenseNetV3[15], and MobileNetV3 [16]. However, it is essential to acknowledge that creating a network architecture is a meticulous and time-consuming process that demands much commitment and effort. Numerous architectural designs have been meticulously crafted to tackle the intricacies of distinct obstacles. The summary of the networks is presented in Table 4.3.

4.3.5 CNN-based Model

To achieve the better accuracy of the proposed model for polyp classification, we have tried different CNN architectures. These architectures are shown in Table 4.4. This task is performed to justify how changing the CNN architecture impacts the model's accuracy.

Table 4.4 Proposed Model Accuracy Table after augmentation 96 x 96 pixel

Model	**CNN Layers**	**Dense Layers**		**Train Accuracy**	**Test Accuracy**
		Neurons	**Dropout**		
NewNet1	16-32-16	128	0.2	0.7803	0.78
NewNet2	16-32-16	128	0.3	0.7628	0.755
NewNet3	16-32-16	128	0.4	0.7491	0.7575
NewNet4	16-32-16	128-64-32	0.2	0.7531	0.7275
NewNet5	32-64-32	128	0.3	0.7934	0.775
NewNet6	32-64-32	128-64-32	0.3	0.7247	0.7075
NewNet7	16-32-16	128-64-32	0.3	0.7734	0.775
NewNet8	16-32-16	128-64-32	0.3	0.715	0.7075
NewNet9	**32-64-32**	**128-64-32**	**0.3**	**0.8119**	**0.785**
NewNet10	32-64-32	128-64-32	0.5	0.6544	0.6775

According to Table 4.4, it is revealed that NewNet9 performs better with respect to the other networks with respect to train accuracy and test accuracy. As this model performs better with respect to other networks, we assume that this network can classify the polyp from the colonoscopy image better with respect to the other network. The NewNet9 model is explained separately at Fig. 4.4.

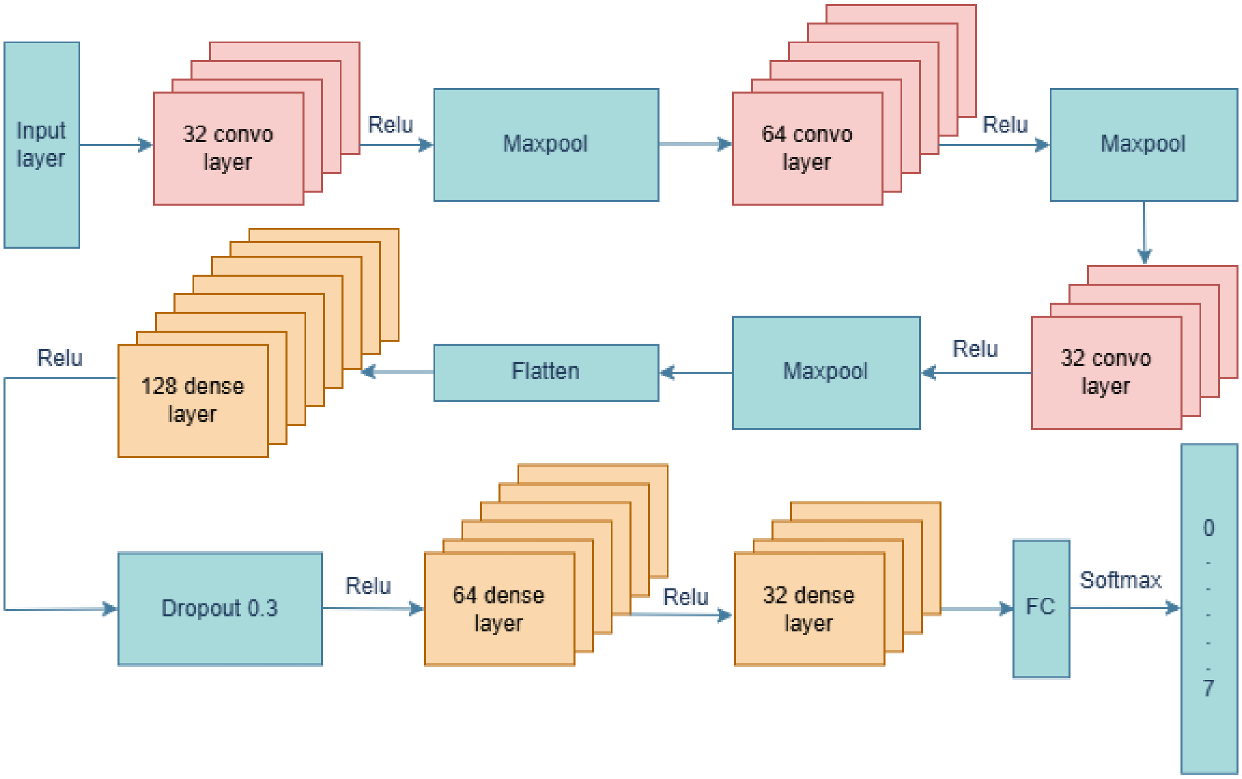


Fig. 4.4 Structure of NewNet9.

Three convolutional layers and three dense layers make to the architecture of the model. In order to effectively extract features from images, we created this network. The hands' characteristics were extracted using a 2-steam CNN model in our base paper [1]. In comparison to the base model, our model is a little bit simpler and performs somewhat better.

We included convolution, max-pooling, rectified linear units (ReLU), dense layering, and dropout in our suggested model. To categorize the polyp, we employed the 4K polyp dataset in our suggested model. We also use the polpy image dataset to test our model. Fig. 4.3 shows the process flowchart for the suggested polyp classification model. Fig. 4.4 provides a visual illustration of our suggested paradigm.

We have 32 filters in our first convolutional layer, each with a kernel size of 3. Additionally, we used max pooling with a 2-kernel size. The 32 filters combine the input image sizes of 96 by 96. We employed the activation function after convolving the input data to aid the network in understanding the intricate pattern of the data. Since Rectified Linear Unit (ReLU) activation is the most widely utilized activation function in the current state of the art, we employed it in our study. This activation function converts any negative number to zero.

To lower the feature map’s dimensionality, we employed max pooling. Two is the maximum pooling filter size.1 stride size was utilized with this maximum pooling. By using the weights’ square values in the cost function, it aids in the reduction of the overfitting issue.

64 filters were implemented in the convolutional neural network’s second layer. Here, we also applied the ReLU activation function to our feature map since we also needed to utilize an activation function. In this layer, we reduced the dimensionality of our feature map by using a max pooling with filter size 2 and stride size 1. Following the use of max pooling, we obtained image data with a size of 16.

Unlike our first convolutional neural networks, we employed 32 filters in the third layer of the neural network. The image from the second convolutional neural network layer, which is 16 layers deep, is convolved by these 64 filters. after using three kernels to convolve this image. Here, we also used the ReLU activation function on our feature map as an activation function. In order to decrease the dimensionality of our feature map, we once more employed max pooling in this layer with filter size 2 and stride size 1. We employed L2 regularization with a 0.001 learning rate to lessen overfitting.

Table 4.5 Hyper-parameters used in the proposed best-selected model are presented

Layer	Activation Function	Learning Rate	Filter Size	Pooling	Batch Size	Optimizer & Loss Function	Dropout
First CNN Layer	ReLU	0.001	32 filters, kernel size 3	Max	32	Adam, Categorial Crossentropy	No
Second CNN Layer	ReLU		64 filters, kernel size 3	Max			No
Third CNN Layer	ReLU		32 filters, kernel size 3	Max			No
FC-1	ReLU		128 Neurons	-			Yes (0.3)
FC-2	ReLU		64 Neurons	-			No
FC-3	ReLU		32 Neurons	-			No
FC-4	ReLU		8 Output Neurons	-			No

We used flatten layer after the convolutional layer. The convolutional layer’s Flatten() function converts the multidimensional output into a one-dimensional array. This is frequently carried out before to the data being sent into a fully linked or densely connected layer.

Following that, a fully connected layer of 128 neurons (or units) is added. Rectified Linear Unit (ReLU), a popular activation function that produces zero for negative values and the input for positive values, is the activation function that is employed. We used dropout after the dense layers. This helped us lessen the overfitting issue we were having. A dropout value of 0.3 was employed. The final layer is a fully connected layer with 8 neurons, representing the output classes (assuming a classification task with 8 classes). The activation function used is softmax, which is commonly used in multi-class classification problems. It converts the raw output scores into probabilities, and the class with the highest probability is predicted as the final output. The Hyper-parameters used in the proposed best selected model are presented in Table. 4.5.

4.3.6 PolypNet: Proposed CNN Model Including Self-Attention Mechanism

The self-attention mechanism significantly enhances the model's capacity to identify and concentrate on the most informative features within complex data. This is particularly beneficial for specialized datasets, where the relevance of specific characteristics can vary significantly across different classification tasks. By allowing the model to dynamically weigh the importance of different elements, self-attention improves feature discrimination.

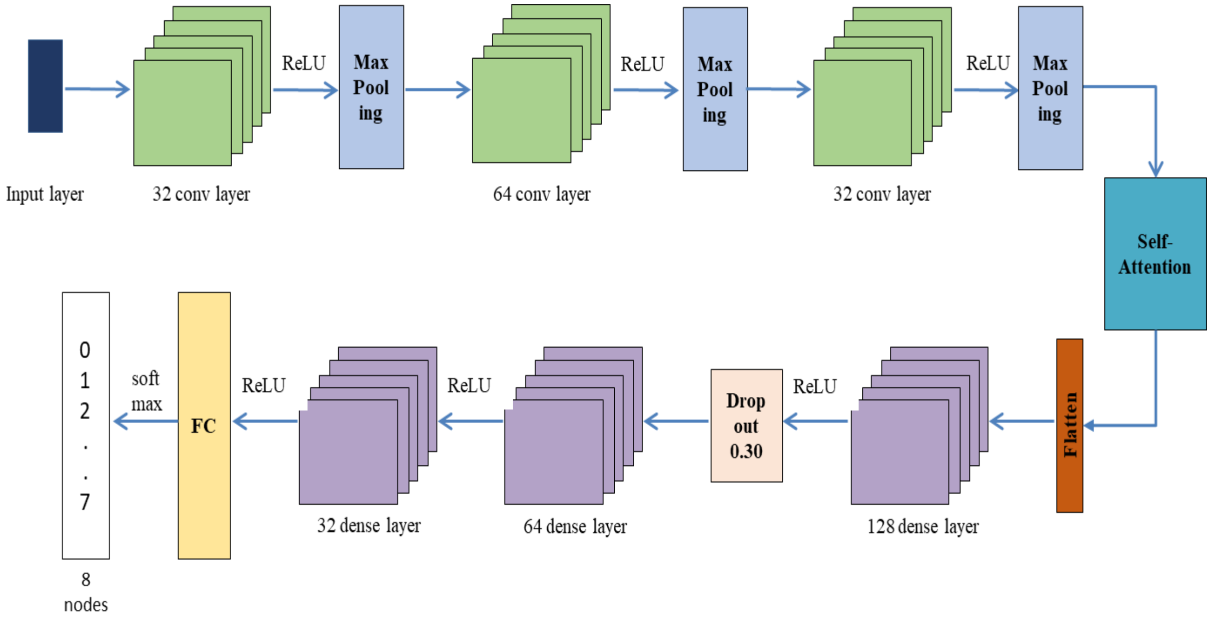


Fig. 4.5 Architectural design of the proposed PolypNet model

In this work, we integrate a self-attention mechanism into a neural network architecture to leverage this capability. Following the initial feature extraction via convolutional layers, the self-attention module assesses the inter-dependencies between features, refining the representation. The architecture then transitions to a sequence of fully connected (dense) layers for final classification. This includes a 32-node layer, a 64-node layer with a dropout rate of 0.3 to prevent overfitting, and culminates in a 128-node output layer. A softmax activation function is typically applied to this final layer for multi-class classification. The complete architectural design of the proposed model, named PolypNet, is illustrated in Fig. 4.5.

4.4 Model Training

We use those pre-trained state-of-the-art models to detect and classify polyps in our input images. Here, all of those CNN models are pre-trained on the ImageNet dataset.

4.4.1 Optimization Algorithm

We use different optimization algorithms such as Adam, RMSprop and SGD to repeatedly adjust the model's parameters during training and reduce the estimated loss. Adam (Adaptive Moment Estimation) combines the advantages of AdaGrad and RMSprop, making it well-suited for handling sparse gradients and noisy data. Adam updates the model using the gradients' first and second instances. These updates were calculated as:

$$M_t = B_1 M_{t-1} + (1 - B_1)gt \tag{4.1}$$

$$V_t = B_2 V_{t-1} + (1 - B_2)g^2 t \tag{4.2}$$

$$\frac{M_t}{1 - B_1} = \frac{1}{V_t(1 - B_2)} = \theta\epsilon \tag{4.3}$$

$$\theta = \Theta_{t-1} - \eta \tag{4.4}$$

$$m_t = A\sqrt{\frac{V_t}{\in} + E} \tag{4.5}$$

4.4.2 Loss Function

We used the loss function "categorical cross-entropy," which is frequently employed, particularly in the context of classification problems. It calculates the discrepancy between the target classes' actual distribution and the neural network's anticipated distribution.

Categorical cross-entropy measures how closely the projected probabilities match the real distribution of class labels in the context of a classification issue where the objective is to allocate an input to one of many predetermined classes.

The formula for categorical cross-entropy is as follows:

$$H(y, \hat{y}) = -\sum_{i} y_i . \log(\hat{y}_i) \qquad (4.6)$$

Where:

- $H(y, \hat{y})$ is the categorical cross-entropy loss.
- y_i is the true probability of class i.
- $\hat{y}_i$ is the predicted probability of class i.

The sum is taken over all classes.

- **Logarithmic Scale**: The use of the logarithm (log) penalizes the model more when it makes predictions that are confidently wrong. The greater the difference between the true and predicted probabilities, the higher the loss.
- **Negative Log-Likelihood**: The formula resembles the negative log-likelihood, emphasizing the idea of maximizing the log-likelihood of the true class. Minimizing the negative log-likelihood is equivalent to minimizing the cross-entropy.
- **Multi-Class Classification**: Categorical cross-entropy is particularly suitable for multi-class classification problems, where each input can belong to one and only one class.
- **Softmax Activation**: Often, the final layer of a neural network for classification tasks is equipped with a softmax activation function. The softmax function converts raw scores (logits) into a probability distribution over classes, and categorical cross-entropy is then applied to compare this distribution with the true labels.
- **In the context of model training**: The goal is to minimize the categorical cross-entropy loss during optimization. This is typically achieved through gradient-based optimization algorithms like stochastic gradient descent (SGD) or its variants.

4.4.3 Training Procedure

To optimize the effectiveness of the proposed method, during training, certain parameters were taken. The batch size, which is set to 32, determines the number of instances handled at once.

Table 4.6 The Variation for Model

Parameters	**Value**
Epoch	100
Batch Size	32
Optimization Algorithm	Adam

Learning rate	0.001
Loss function	categorical cross-entropy

Additionally, we select 100 as the maximum number of epochs. The learning rate acts as a step size for the model to modify its output. We train the model using the Adam optimization function and a learning rate 0.0001. We train the model and evaluate the result using the following variations listed in the Table 4.6.

4.5 Summary

The dataset for detecting and classifying polyps from images is publicly available and balanced. Here, we used VGG16, ResNet50, DenseNetv3, and MobileNetV3, and we also developed some new CNN models and discovered the best one among them. We used our dataset without manipulation and tried to find the best output.

5. CHAPTER 5: EXPERIMENTS AND RESULTS

5.1 Overview

This segment summarizes the findings of our study, " PolypNet: An Attention Based Shallow Deep Learning Model for Colorectal Polyp Image Classification". We detail the experimental setup, including the dataset, preprocessing methods, and the proposed framework. Our approach evaluates several pre-trained state-of-the-art models—VGG16, ResNet50, DenseNetV3, and MobileNetV3—on the Kvasir dataset. We also introduce PolypNet, a novel shallow CNN model. Through in-depth analysis and comparison, we demonstrate the superiority of our proposed strategy over existing approaches.

5.2 Datasets and Experimental Settings

As detailed in Chapter 4 (Methodology, Section 4.2: Datasets), the Kvasir-v1 dataset was used for model training and evaluation. The dataset was partitioned into training, testing, and validation sets using an 80:10:10 ratio.

5.3 Experimental Tools and Environment

The experiments were carried out on Google Collaboratory, utilizing a GPU runtime infrastructure equipped with 12.7 GB of System RAM, 15.0 GB of GPU RAM, and 78.2 GB of Disk capacity. Python 3.6, with the inclusion of Keras integrated into TensorFlow, streamlined the implementation of machine learning models. The architectural designs of the DL models were acquired from publicly accessible web sources. Specific hyperparameters were fine-tuned and optimized to improve the designs. The settings used were batch sizes of 32, a learning rate of 0.001, and 200 epochs of training.

5.4 Evaluation Matrix

The models' classification performance for a given set of test data is evaluated using a matrix called the confusion matrix. The two axes of the matrix show the true and anticipated values as well as the total number of forecasts [11]. We do a number of computations for our model using this matrix.

- **Confusion Matrix (for classification problems)**: A table used to evaluate the performance of a classification algorithm, breaking down true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

- **Accuracy**: Accuracy is a fundamental metric for evaluating classification models, representing the overall proportion of correct predictions made across all classes. It is calculated as the ratio of the number of correct predictions—comprising both True Positives (TP) and True Negatives (TN)—to the total number of predictions, which includes TP, TN, False Positives (FP), and False Negatives (FN). The formula for accuracy is given by Equation (5.1):

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{5.1}$$

- **Precision**: A model's precision is determined by how well it predicts positive results. The calculation involves determining the proportion of correctly identified positive results (true positives) from the overall number of positive predictions (true positives and false positives). Precision gives us an understanding of the model's capacity to correctly identify adverse events without misclassifying them as positive. The precision equation is presented in Equation (5.2).

$$Precision = \frac{TP}{(TP + FP)} \tag{5.2}$$

- **Recall**: Recall, also known as sensitivity, measures the model's ability to identify positive instances correctly. It's calculated by dividing the number of true positives by the sum of true positives and false negatives. The recall equation is presented in Equation (5.3).

$$Recall = \frac{TP}{(TP + FN)} \tag{5.3}$$

- **F1 Score**: The F1 Score is calculated as the average of accuracy and recall. It offers a single score that combines recall and accuracy, achieving a balance between the two measures. The F1 Score equation is presented in Equation (5.4).

$$F1\, Score = 2\, \frac{(Precision \times Recall)}{(Precision + Recall)} \tag{5.4}$$

5.5 Experimental Results

Our results are divided into three parts. The initial evaluation involves benchmarking the dataset against several state-of-the-art transfer learning models. Subsequently, we analyze a series of ten custom CNN models (NewNet1 to NewNet10) to determine the most effective architecture. From this comparison, NewNet9 was chosen as our proposed model due to its superior accuracy of 0.785, even though this result remains relatively low. In the final phase, we augment the best-performing model, NewNet9, with a self-attention mechanism to maximize classification accuracy.

5.5.1 Transfer Learning

We used VGG16, ResNet50, DenseNetv3, and MobileNetv3 on our dataset for classification of polyps, and among those, RestNet50 performed best and MobileNetv3 worst. The Table 5.1 represents the training and testing results:

Table 5.1 Accuracy of state-of-the-art CNN models

Model Name	**Train Accuracy**	**Test Accuracy**
VGG16	1	0.84
ResNet50	1	0.86
DenseNetv3	0.9828	0.74
MobileNetv3	1	0.69

5.5.1.1 VGG16

The training behavior of the VGG16 model on the Kvasir dataset is illustrated in Figures 5.1 and 5.2. As shown in the loss curve (Fig. 5.1), the training loss begins at a high value before undergoing a substantial decline and eventual stabilization, which signifies effective learning. However, the validation loss curve initially mirrors this trend but subsequently begins to oscillate, a clear indicator of model overfitting. This is further corroborated by the accuracy plot in Fig. 5.2, where the training accuracy rapidly reaches and maintains a near-perfect value of 1.00, while the validation accuracy plateaus at a lower value of approximately 0.80 with minor fluctuations. The final classification performance of the VGG16 model is detailed in the confusion matrix presented in Fig. 5.3.

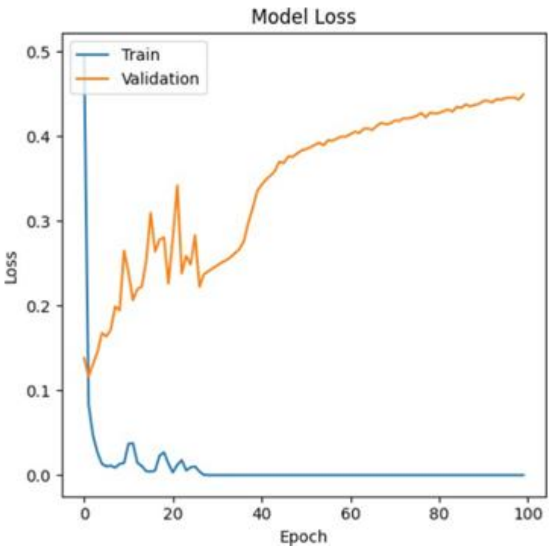


Fig. 5.1 Loss Curve of VGG16.

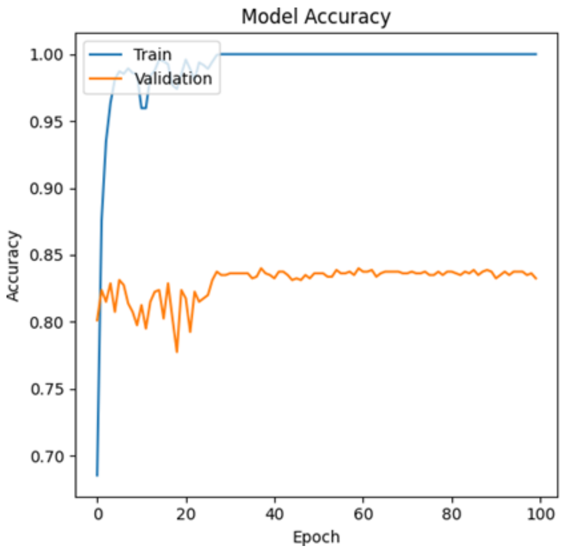
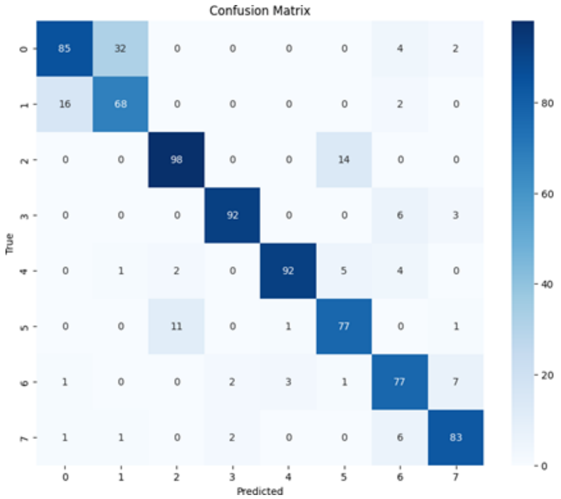


Fig. 5.2 Accuracy Curve of VGG16.



	0	1	2	3	4	5	6	7
0	85	32	0	0	0	0	4	2
1	16	68	0	0	0	0	2	0
2	0	0	98	0	0	14	0	0
3	0	0	0	92	0	0	6	3
4	0	1	2	0	92	5	4	0
5	0	0	11	0	1	77	0	1
6	1	0	0	2	3	1	77	7
7	1	1	0	2	0	0	6	83

Fig. 5.3 Confusion Matrix of VGG16.

5.5.1.2 ResNet50

The performance of the ResNet50 model is detailed across Figures 5.4, 5.5, and 5.6. The loss curve in Fig. 5.4 reveals a training loss that begins at 0.40 and decreases rapidly, stabilizing near zero, while the validation loss follows an opposite trajectory, increasing after the 10th epoch and indicating potential overfitting. This divergence is mirrored in the accuracy curve (Fig. 5.5), where the training accuracy quickly approaches perfection, but the validation accuracy stabilizes at a lower, fluctuating range of 0.85-0.90, failing to match the training performance. The resultant classification outcomes of the model are further elaborated in the confusion matrix presented in Fig. 5.6.

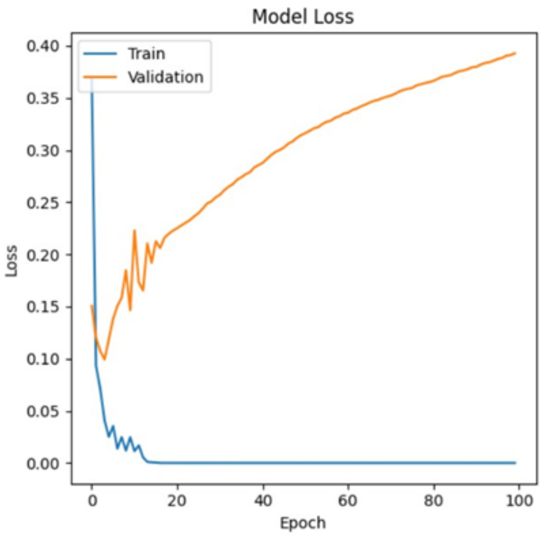


Fig. 5.4 Loss Curve of ResNet50.

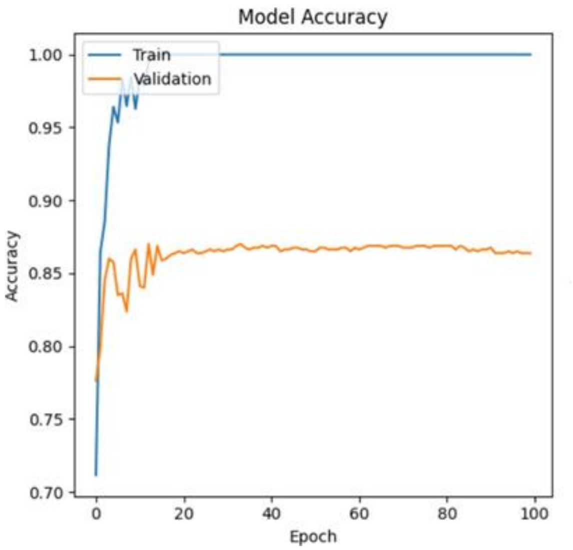


Fig. 5.5 Accuracy Curve of ResNet50.

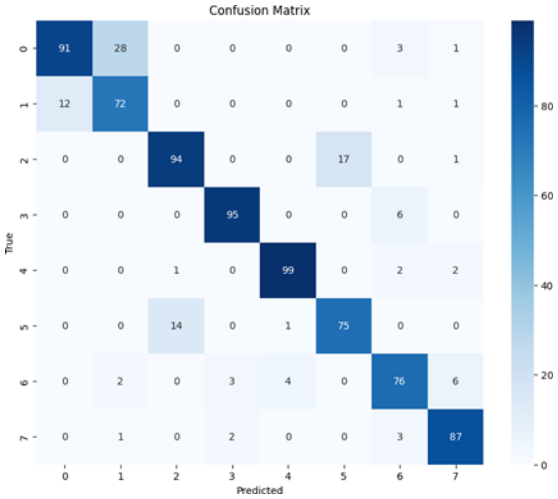


Fig. 5.6 Confusion Matrix of ResNet50

5.5.1.3 DenseNetv3

The training dynamics of the DenseNetV3 model, as visualized in Figures 5.7 and 5.8, indicate a significant performance discrepancy between the training and validation phases. Figure 5.7 shows the training loss converging rapidly to near zero, while the validation loss remains volatile and fails to follow this trend, signaling potential overfitting after approximately 60 epochs. This is further evidenced in the accuracy plot (Fig. 5.8), where the training accuracy shows a consistent rise, but the validation accuracy struggles with fluctuations before plateauing at a modest 0.75, underscoring the model's poor generalization to unseen data. The final classification results of the DenseNetV3 model are captured in the confusion matrix in Fig. 5.9.

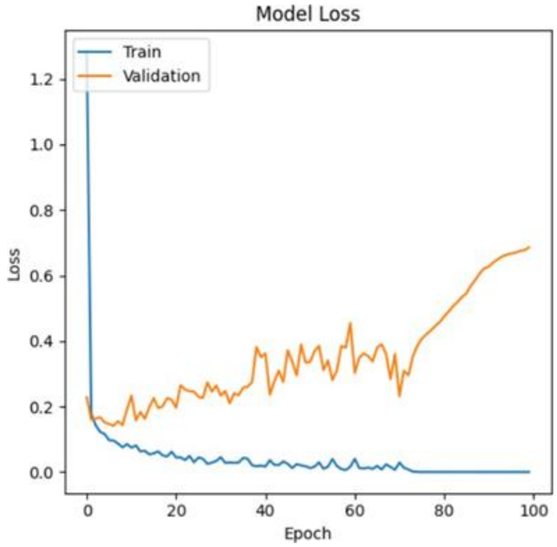


Fig. 5.7 Loss Curve of DenseNetV3.

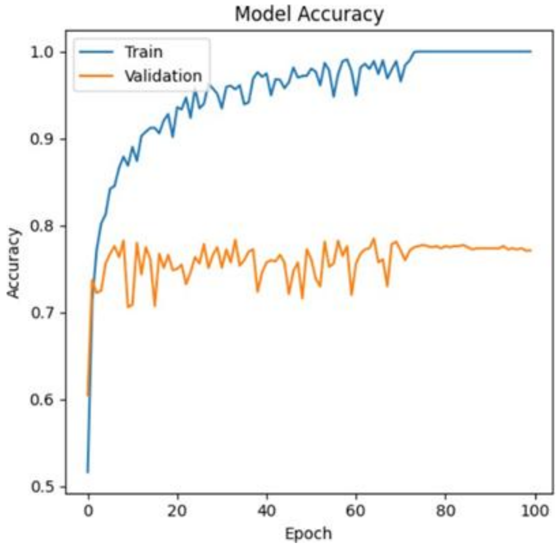


Fig. 5.8 Accuracy Curve of DenseNetv3.

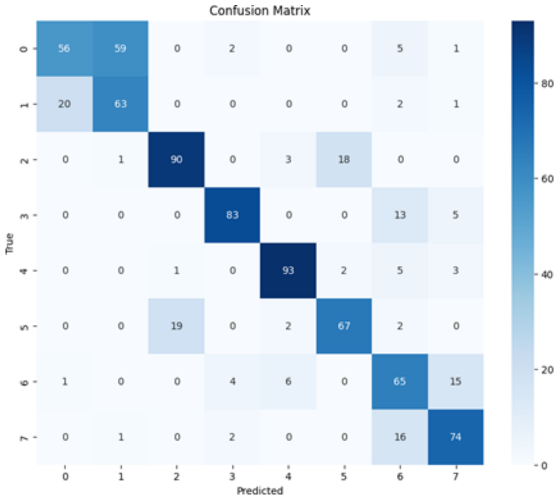


Fig. 5.9 Confusion matrix of DenseNetV3.

5.5.1.4 MobileNetv3

The performance of the MobileNetV3 model is detailed in Figures 5.10 through 5.12. The loss curve in Fig. 5.10 indicates effective learning, with training loss rapidly decreasing from 1.2, while the fluctuating and subsequently rising validation loss after 60 epochs suggests potential overfitting. This divergence is corroborated by the accuracy curve in Fig. 5.11, where the training accuracy quickly converges to near-perfect levels, but the validation accuracy, after initial volatility, plateaus at a significantly lower rate of approximately 0.7. The resultant classification performance of the model is further elaborated in the confusion matrix presented in Fig. 5.12.

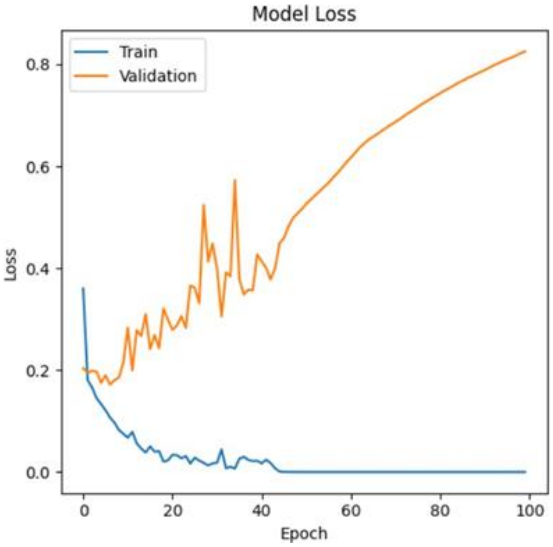


Fig. 5.10 Loss Curve of MobileNetv3.

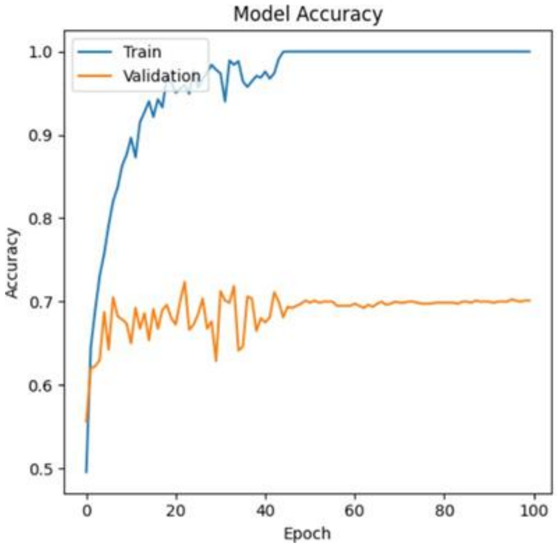


Fig. 5.11 Accuracy Curve of MobileNetv3.

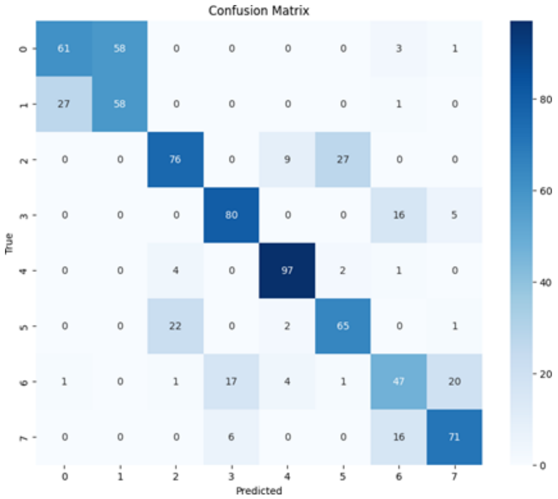


Fig. 5.12 Confusion matrix of MobileNetv3.

5.5.1.5 Experimental analysis of the transfer learning models

The comparison graph in Fig. 5.16 highlights the significant variations in accuracy among state-of-the-art models. With towering training accuracy peaks of 0.84 and 0.86, respectively, VGG16 and ResNet50 stand out and demonstrate their strong learning capabilities. By comparison, MobileNetv3 has the lowest training accuracy, measuring at around 0.69, suggesting that there are comparatively more obstacles to successfully collecting training data patterns. These varying peaks of training accuracy reveal the different learning capacities of the models being assessed.

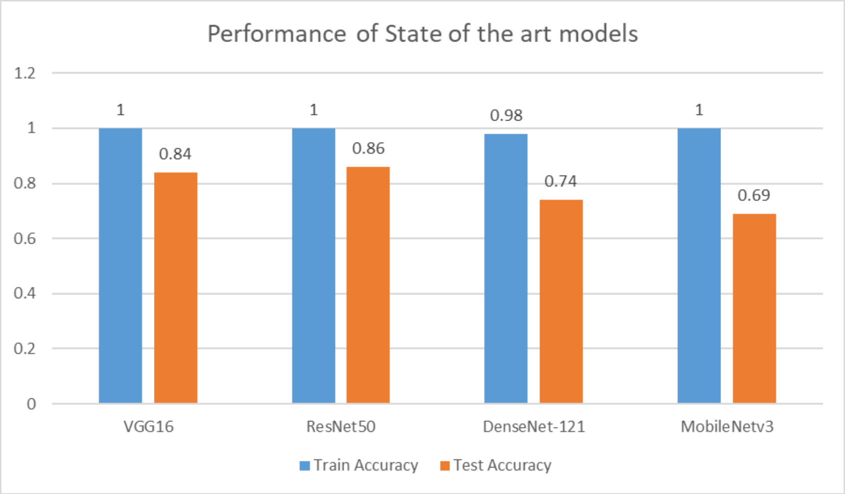


Fig. 5.13 Transfer Learning Model Comparison.

5.5.2 Shallow CNN Model

This section details the development and selection of a custom Shallow CNN model to address the limitations of larger, pre-trained networks. We designed and evaluated ten different shallow architectures (NewNet1 to NewNet10) to identify the optimal structure for polyp classification. From this comparative analysis, we selected a model as the best-performing model.

5.5.2.1 Selection of CNN Model

To achieve the best CNN models, this section proposes ten separate CNN models. Based on the accuracy and regularized performance of the model, the best model is selected as the proposed model. This section also explained how model accuracy is changed by modifying the model structure. The Table 5.2 that follows represents our proposed model. Ten convolutional neural network (CNN) models are included in the table, along with information on their architecture, number of layers, dense layers, and parameters. Every row is a different model that illustrates structural differences. revealing information on how effective each is in comparison to a certain activity.

Table 5.2 Proposed Model and its architecture for the augmented dataset of 96 x 96

Model	CNN layers	Dense Layers		Parameters	Train Accuracy	Test Accuracy
		Neurons	Dropout			
NewNet1	16-32-16	128	0.2	448-4640-4624	0.7803	0.78
NewNet2	16-32-16	128	0.3	448-4640-4624	0.7628	0.755
NewNet3	16-32-16	128	0.4	448-4640-4624	0.7491	0.7575
NewNet4	16-32-16	128-64-32	0.2	448-4640-4624	0.7531	0.7275
NewNet5	32-64-32	128	0.3	896-18496-18464	0.7934	0.775
NewNet6	32-64-32	128-64-32	0.3	896-18496-18464	0.7247	0.7075
NewNet7	16-32-16	128-64-32	0.3	448-4640-4624	0.7734	0.775
NewNet8	16-32-16	128-64-32	0.3	448-4640-4624	0.715	0.7075
NewNet9	**32-64-32**	**128-64-32**	**0.3**	**896-18496-18464**	**0.8119**	**0.785**
NewNet10	32-64-32	128-64-32	0.5	896-18496-18464	0.6544	0.6775

The NewNet9 is our selected proposed CNN model as it shows the best performance, according to train and test accuracy.

5.5.2.2 Experimental analysis of the CNN Model

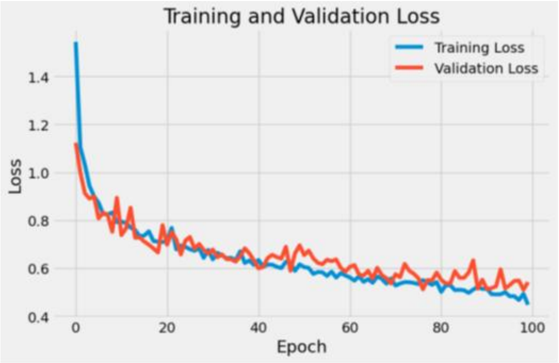


Fig. 5.14 Loss curve of the proposed CNN model

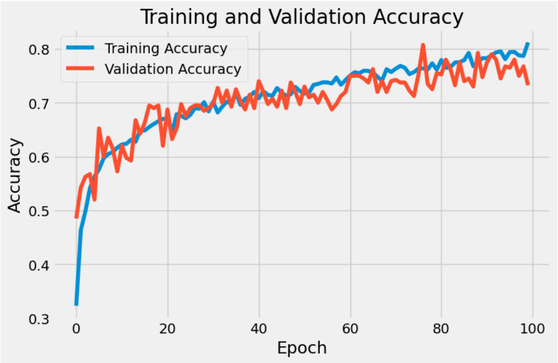
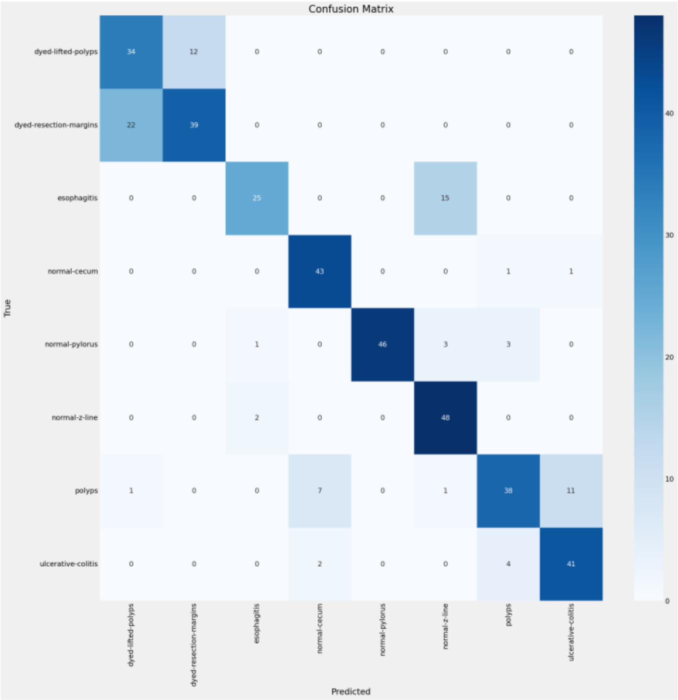


Fig. 5.15 Accuracy curve of the proposed CNN model

The proposed CNN model was evaluated on the augmented Kvasir-v1 dataset with images resized to 96x96 pixels. As illustrated in the loss and accuracy curves (Figs. 5.14 and 5.15, respectively), the model achieved a final accuracy of 0.78, corresponding to a misclassification rate of 0.22 after 100 epochs. The confusion matrix in Fig. 5.16 provides a detailed breakdown of this performance. However, this level of precision was deemed unsatisfactory when compared to state-of-the-art transfer learning models. To address this limitation and enhance the model's accuracy, self-attention mechanisms were integrated into the architecture, resulting in the final model which is presented in the following section.



	dyed-lifted-polyps	dyed-resection-margins	esophagitis	normal-cecum	normal-pylorus	normal-z-line	polyps	ulcerative-colitis
dyed-lifted-polyps	34	12	0	0	0	0	0	0
dyed-resection-margins	22	39	0	0	0	0	0	0
esophagitis	0	0	25	0	0	15	0	0
normal-cecum	0	0	0	43	0	0	1	1
normal-pylorus	0	0	1	0	46	3	3	0
normal-z-line	0	0	2	0	0	48	0	0
polyps	1	0	0	7	0	1	38	11
ulcerative-colitis	0	0	0	2	0	0	4	41

Fig. 5.16 Confusion Matrix of the proposed CNN model

5.5.3 PolypNet: Shallow CNN Model Including Self-Attention Mechanisms

In order to solve the noted accuracy issues, we proposed an improved version of the shallow CNN Model in this section that includes attention methods. The study of the experimental data, which showed that the accuracy of the original shallow CNN model was not up to par, led to the decision to incorporate attention processes. Attention methods are utilized to enhance the model's capacity to understand complex interdependencies and connections within the data. This section describes the changes made to the design, explains the reasoning behind the addition of attention mechanisms, and illustrates how these improvements help achieve higher accuracy. The final CNN model, which is now furnished with attention mechanisms, is detailed, highlighting its ability to surmount the obstacles found in the first shallow CNN Model performance study.

5.5.3.1 PolypNet Without Augmentation

The initial performance of the improved shallow CNN model, trained on the non-augmented main dataset, is visualized in Figures 5.17 to 5.19. The loss curve in Fig. 5.17 indicates a clear case of overfitting, where the model fails to generalize effectively. This is further substantiated by the accuracy curve in Fig. 5.18, which likely shows a growing disparity between high training accuracy and lower validation accuracy. The specific classification results of this initial model are detailed in the confusion matrix presented in Fig. 5.19.

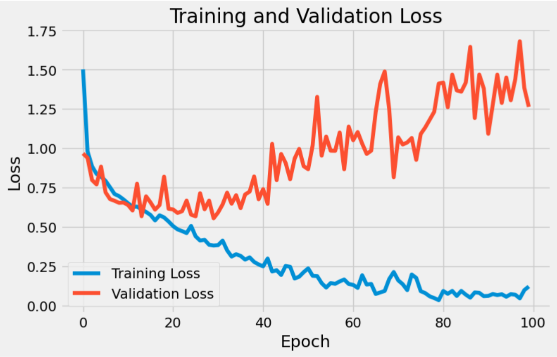


Fig. 5.17 Loss curve of PolypNet to the main dataset

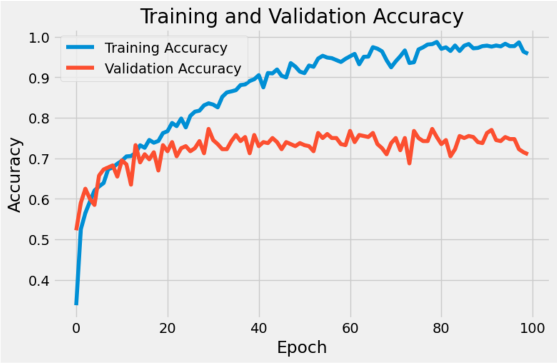
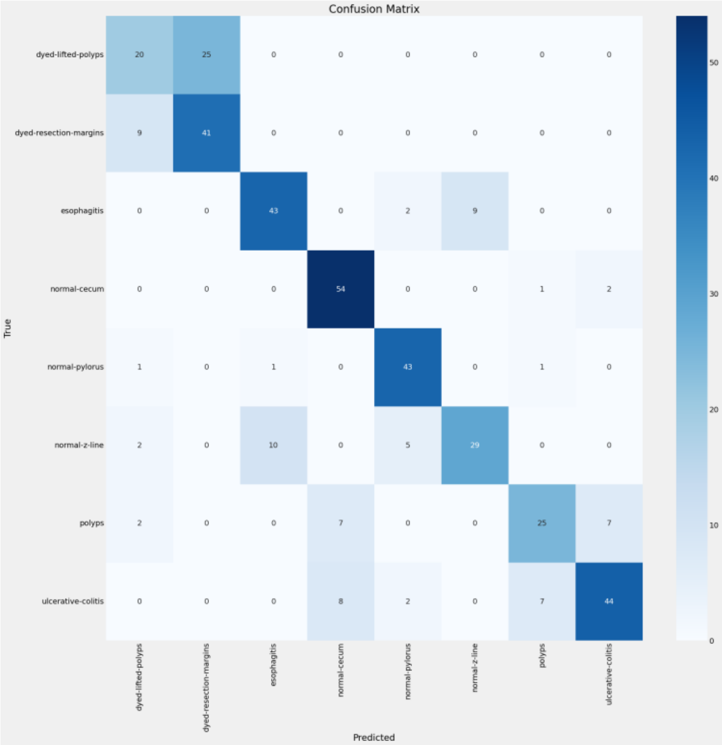


Fig. 5.18 Accuracy curve of PolypNet to the main dataset



	dyed-lifted-polyps	dyed-resection-margins	esophagitis	normal-cecum	normal-pylorus	normal-z-line	polyps	ulcerative-colitis
dyed-lifted-polyps	20	25	0	0	0	0	0	0
dyed-resection-margins	9	41	0	0	0	0	0	0
esophagitis	0	0	43	0	2	9	0	0
normal-cecum	0	0	0	54	0	0	1	2
normal-pylorus	1	0	1	0	43	0	1	0
normal-z-line	2	0	10	0	5	29	0	0
polyps	2	0	0	7	0	0	25	7
ulcerative-colitis	0	0	0	8	2	0	7	44

Fig. 5.19 Confusion Matrix of PolypNet on the main dataset

5.5.3.2 PolypNet With Augmentation

To mitigate the overfitting problem observed in the initial model, the improved shallow CNN was retrained on an augmented dataset. The subsequent loss curve, depicted in Fig. 5.20, demonstrates a more desirable convergence pattern between the training and validation sets. This improved generalization is further reflected in the accuracy curve shown in Fig. 5.21, where the validation performance more closely tracks the training accuracy. The final classification outcomes of this refined model, trained with data augmentation, are detailed in the confusion matrix presented in Fig. 5.22.

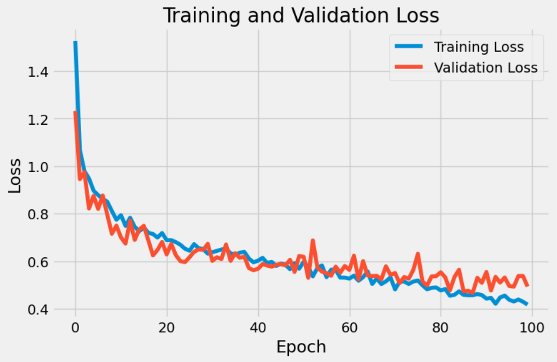


Fig. 5.20 Loss curve of PolypNet with augmentation

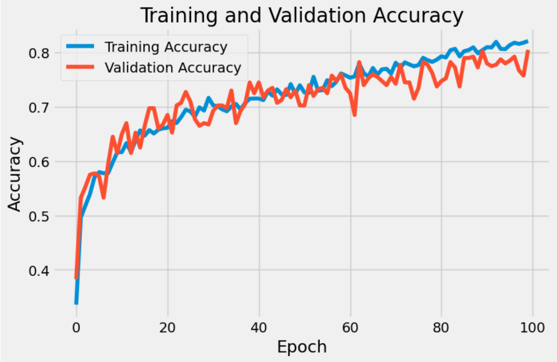


Fig. 5.21 Accuracy curve of PolypNet with augmentation

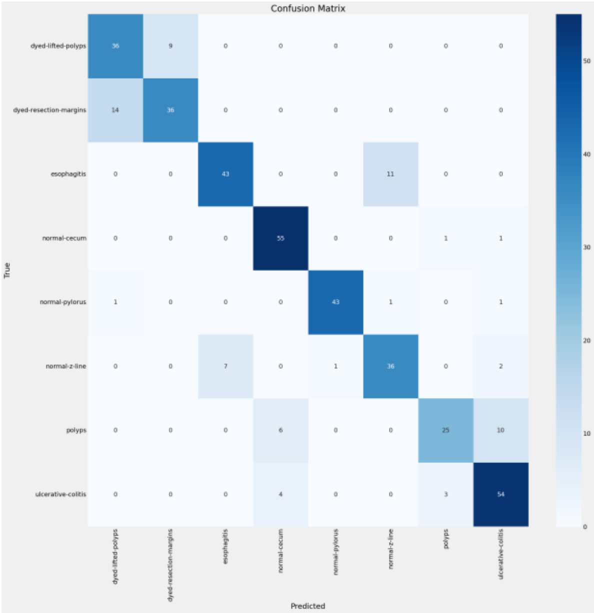


Fig. 5.22 Confusion matrix of PolypNet with augmentation

5.5.3.3 PolypNet For 200 Epoch

As the model failed to converge and exhibited high loss after 100 epochs, the proposed PolypNet was trained for an extended 200 epochs on the augmented dataset to enhance accuracy and reduce the loss. The resulting loss curve, presented in Fig. 5.23, confirms a more stable convergence. This improvement is further validated by the accuracy curve in Fig. 5.24, which shows enhanced learning dynamics. The final classification performance achieved by this optimized model is detailed in the confusion matrix provided in Fig. 5.25.

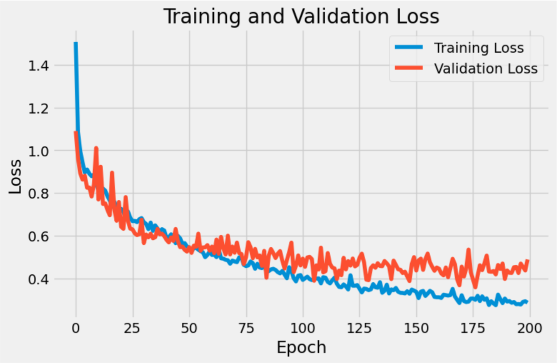


Fig. 5.23 Loss Curve of PolypNet with Augmentation for 200 epoch

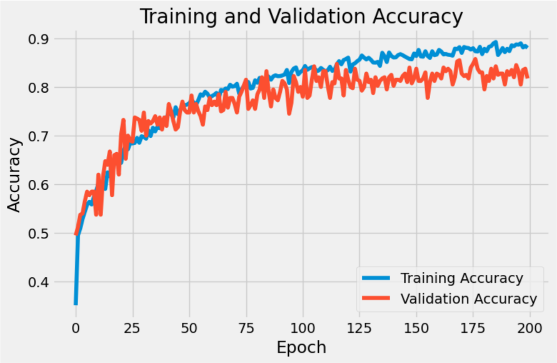


Fig. 5.24 Accuracy Curve of PolyNet With augmentation for 200 epoch

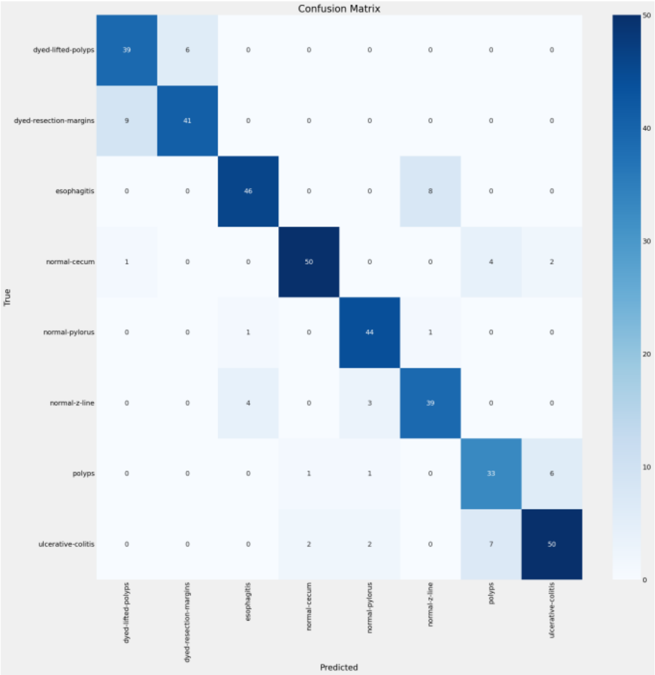


Fig. 5.25 Confusion Matrix of PolypNet With augmentation for 200 epoch

5.6 Comparison & Discussion

The experimental result showed that the PolypNet model performed better than a few state-of-the-art transfer learning models to accurately classify colorectal polyps. Table 5.3 and Table 5.4 show the results, which show that the proposed CNN method is better than VGG16, DenseNetv3, and MobileNetv3. It got an accuracy score and F1-score of 0.86 and performed almost as well as

ResNet50. The success of the proposed model is largely attributed to its focus on architectural design and the preprocessing techniques applied to the dataset.

Table 5.3 The Training, Validation and Test Accuracy of Polypnet, and TL Models

Model	**Train Accuracy (%)**	**Validation Accuracy (%)**	**Test Accuracy (%)**
VGG16	100.00	83.00	84.00
ResNet50	100.00	86.00	86.00
DesnseNet-v3	98.00	77.00	74.00
MobileNetv3	100.00	70.00	69.00
PolypNet (100 epochs)	82.00	82.00	82.00
PolypNet (200 epochs)	**88.00**	**82.00**	**86.00**

Table 5.4 The Precision, Recall, And F1-Score of Polypnet, And Transfer Learning Models

Model	**Precision**	**Recall**	**F1-Score**
VGG16	0.85	0.84	0.84
ResNet50	0.87	0.86	0.86
DenseNetv3	0.75	0.74	0.74
MobileNetv3	0.70	0.69	0.69
PolypNet	**0.86**	**0.85**	**0.86**

Finally, Table 5.5 presents a comparison with the current methods for classifying colorectal polyp with the Kvasir-v1 dataset. It is evident that the suggested strategy works better than cutting-edge approaches and that accuracy has significantly improved.

Table 5.5 The Experiment Comparison with Few Current Methods

Method	**Precision**	**Recall**	**F1 Score**
MedMamba [22]	78.74	78.83	78.56
HiFuse [23]	85.08	85.00	84.96
PolypNet	**86.00**	**85.00**	**86.00**

5.7 Ablation Study

The proposed PolypNet method is a combination of shallow CNN and self-attention mechanisms. To justify the effectiveness of the proposed PolypNet, we compare the proposed model with the different components. Initially, the model is tested only using the shallow CNN layers and achieves an accuracy of 78.00%. After that, the self-attention mechanism is incorporated with the shallow CNN layers. This addition shows the improvement of the method, which is an accuracy of 82.00%. That is a 4.00% improvement over the previous method. However, this is performed for the 100 epochs. To justify whether the epoch size has any impact on the model's accuracy, we train our proposed model (PolypNet) for the 200 epochs. This configuration shows significant improvement of the proposed method with an accuracy of 86.00%, which is the 8% and 4% improvement of shallow CNN and shallow CNN+self-attention with 100 epochs, respectively. The result is shown in Table 5.6.

Table 5.6 The Experiment Result of Different Component

CNN	Self-attention	Epoch	Accuracy
√	×	100	78.00%
√	√	100	82.00%
√	√	200	86.00%

6. CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this work, we introduced a shallow CNN-based deep learning model to overcome limitation of overfilling occurs on the small dataset. That will help to classify colorectal polyps accurately with generalized form, contributing to the fight against colorectal polyp cancer effectively. Our model, built with a shallow CNN structure combined with self-attention, shows effective performance. Through comprehensive analysis, our solution consistently outperforms a few existing polyp classification approaches by comparing various DL models. Experiments on the open-source Kvasir-v1 dataset validate the model's effectiveness. Notably, it surpasses several existing deep learning models, achieving near-perfect F1-scores.

6.2 Contribution of the work

The contributions of this paper are summarized below:

1. Introduction of PolypNet, a shallow CNN model with a self-attention mechanism for improved colorectal polyp classification and overcoming the overfitting problem.
2. Comparison of PolypNet with established CNN-based models (VGG16, ResNet50, DenseNetv3, MobileNetv3) to justify the effectiveness of the proposed method.
3. PolypNet achieves a precision score of 0.86, matching ResNet50 and outperforming other models, demonstrating its effectiveness for automated polyp classification.

6.3 Future Work

There are certainly scopes of improvement in our model. In the future, we will work on creating hybrid model architectures that combine several deep learning models to use their strengths and improve diagnostic accuracy across a range of datasets, such as Medmnist2D. Furthermore, making applications for this automated categorization method easy for people to use could also make it easier to use in clinical settings, especially in rural or underdeveloped areas. This would make polyp diagnosis and treatment more accessible.

Outcomes in terms of Research

Paper Title: **PolypNet: An Attention Based Shallow Deep Learning Model for Colorectal Polyp Image Classification**

DOI: 10.1109/ICCIT64611.2024.11022357

Conference: December 2024, 27th International Conference on Computer and Information Technology (ICCIT) At: Cox’s Bazar, Bangladesh

7. REFERENCES

[1] C. A. Doubeni, D. A. Corley, V. P. Quinn, C. D. Jensen, A. G. Zauber, M. Goodman, J. R. Johnson, S. J. Mehta, T. A. Becerra, W. K. Zhao, et al., "Effectiveness of screening colonoscopy in reducing the risk of death from right and left colon cancer: a large community-based study," Gut, vol. 67, no. 2, pp. 291–298, 2018.

[2] D. K. Rex, C. R. Boland, J. A. Dominitz, F. M. Giardiello, D. A. Johnson, T. Kaltenbach, T. R. Levin, D. Lieberman, and D. J. Robertson, "Colorectal cancer screening: Recommendations for physicians and patients from the US Multi-Society Task Force on Colorectal Cancer," Gastroenterology, vol. 153, no. 1, pp. 307–323, 2017.

[3] C. Burke, V. Kaul, and H. Pohl, "Polyp resection and removal procedures: Insights from the 2017 Digestive Disease Week," Gastroenterology & Hepatology, vol. 13, Suppl. 2, p. 1, 2017.

[4] K. Li, M. I. Fathan, K. Patel, T. Zhang, C. Zhong, A. Bansal, A. Rastogi, J. S. Wang, and G. Wang, "Colonoscopy polyp detection and classification: Dataset creation and comparative evaluations," PLoS One, vol. 16, no. 8, p. e0255809, 2021.

[5] A. Haj-Manouchehri and H. M. Mohammadi, "Polyp detection using CNNs in colonoscopy video," IET Computer Vision, vol. 14, no. 5, pp. 241–247, 2020.

[6] A. Nogueira-Rodríguez, R. Domínguez-Carbajales, F. Campos-Tato, J. Herrero, M. Puga, D. Remedios, L. Rivas, E. Sánchez, A. Iglesias, J. Cubiella, et al., "Realtime polyp detection model using convolutional neural networks," Neural Computing and Applications, vol. 34, no. 13, pp. 10375–10396, 2022.

[7] J. Nisha, V. P. Gopi, and P. Palanisamy, "Automated colorectal polyp detection based on image enhancement and dual-path CNN architecture," Biomedical Signal Processing and Control, vol. 73, p. 103465, 2022.

[8] K. Yang, S. Chang, Z. Tian, C. Gao, Y. Du, X. Zhang, K. Liu, J. Meng, and L. Xue, "Automatic polyp detection and segmentation using Shuffle Efficient Channel Attention Network," Alexandria Engineering Journal, vol. 61, no. 1, pp. 917–926, 2022.

[9] A. Nogueira-Rodríguez, R. Domínguez-Carbajales, F. Campos-Tato, J. Herrero, M. Puga, D. Remedios, L. Rivas, E. Sánchez, A. Iglesias, J. Cubiella, et al., "Realtime polyp detection model using convolutional neural networks," Neural Computing and Applications, vol. 34, no. 13, pp. 10375–10396, 2022.

[10] S. Mazumdar, S. Sinha, S. Jha, and B. Jagtap, "Computer-aided automated diminutive colonic polyp detection in colonoscopy by using deep machine learning system: First indigenous algorithm developed in India," Indian Journal of Gastroenterology, 2023, pp. 1–7.

[11] M. Bilal, J. R. G. Brown, and T. M. Berzin, "Using computer-aided polyp detection during colonoscopy," ACG Case Reports Journal, vol. 7, no. 7, pp. 963–966, 2020.

[12] C. Sánchez-Montes, J. Bernal, A. García-Rodríguez, H. Córdova, and G. Fernández-Esparrach, "Review of computational methods for the detection and classification of polyps in colonoscopy imaging," Gastroenterología y Hepatología (English Edition), vol. 43, no. 4, pp. 222–232, 2020.

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

[14] V. Blanes-Vidal, G. Baatrup, and E. S. Nadimi, "Addressing priority challenges in the detection and assessment of colorectal polyps from capsule endoscopy and colonoscopy in colorectal cancer screening using machine learning," Acta Oncologica, vol. 58, suppl. 1, pp. S29–S36, 2019.

[15] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 4700–4708.

[16] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, and Q. V. Le, "Searching for MobileNetV3," in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 1314–1324.

[17] Y.-C. Jheng, Y.-P. Wang, H.-E. Lin, K.-Y. Sung, Y.-C. Chu, H.-S. Wang, J.-K. Jiang, M.-C. Hou, F.-Y. Lee, and C.-L. Lu, "A novel machine learning-based algorithm to identify and classify lesions and anatomical landmarks in colonoscopy images," Surg. Endosc., vol. 36, pp. 640–650, 2022.

[18] W. Tavanapong, J. Oh, M. A. Riegler, M. Khaleel, B. Mittal, and P. C. De Groen, "Artificial intelligence for colonoscopy: Past, present, and future," IEEE J. Biomed. Health Inform., vol. 26, no. 8, pp. 3950–3965, 2022.

[19] D. Taha, A. Alzu'bi, A. Abuarqoub, M. Hammoudeh, and M. Elhoseny, "Automated colorectal polyp classification using deep neural networks with colonoscopy images," Int. J. Fuzzy Syst., pp. 1–13, 2021.

[20] J. Y. Lee, J. Jeong, E. M. Song, C. Ha, H. J. Lee, J. E. Koo, D.-H. Yang, N. Kim, and J.-S. Byeon, "Real-time detection of colon polyps during colonoscopy using deep learning: Systematic validation with four independent datasets," Sci. Rep., vol. 10, no. 1, p. 8379, 2020.

[21] K. Pogorelov, K. R. Randel, C. Griwodz, S. L. Eskeland, T. de Lange, D. Johansen, C. Spampinato, D.-T. Dang-Nguyen, M. Lux, P. T. Schmidt, M. Riegler, and P. Halvorsen, "KVASIR: A multi-class image dataset for computer aided gastrointestinal disease detection," in Proc. 8th ACM Multimedia Systems Conf. (MMSys

[22] J. Wang et al. MedMamba: multi-scale deformable attention via state space models for robust medical image segmentation Biomed. Signal Process. Control (2026)

[23] X. Huo, HiFuse: hierarchical multi-scale feature fusion network for medical image classification, Biomed. Signal Process. Control, (2024)

[24] Q. Wang, H. Che, W. Ding, L. Xiang, G. Li, Z. Li, and S. Cui, "Colorectal Polyp Classification from White-light Colonoscopy Images via Domain Alignment," in *Proceedings of the 24th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2021. [Online]. Available: https://arxiv.org/abs/2108.02476

[25] J. F. Lazo et al., "Semi-supervised Bladder Tissue Classification in Multi-Domain Endoscopic Images," IEEE Transactions on Biomedical Engineering, early access, 2023, doi: 10.1109/TBME.2023.3331953. [Online]. Available: https://zenodo.org/record/7741476.

[26] C. Yang, Z. An, L. Cai, and Y. Xu, "Hierarchical Self-supervised Augmented Knowledge Distillation," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 9, pp. 10649-10657, Jun. 2023. [Online]. Available: https://github.com/winycg/HSAKD

[27] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using EfficientNet deep learning model," Ecological Informatics, vol. 61, p. 101182, Mar. 2021, doi: 10.1016/j.ecoinf.2020.101182.

[28] M. Yusuf, A. F. D. Kana, M. A. Bagiwa, and M. Abdullahi, "Multi-classification of breast cancer histopathological image using enhanced shallow convolutional neural network," Scientific African, vol. 21, p. e01812, Sep. 2023, doi: 10.1016/j.sciaf.2023.e01812.

[29] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," Nature, vol. 542, no. 7639, pp. 115–118, Feb. 2017, doi: 10.1038/nature21056.

[30] Y. Ling, Y. Wang, W. Dai, J. Yu, P. Liang, and D. Kong, "MTANet: Multi-Task Attention Network for Automatic Medical Image Segmentation and Classification," IEEE Journal of Biomedical and Health Informatics, vol. 27, no. 9, pp. 4412-4423, Sept. 2023, doi: 10.1109/JBHI.2023.3289941.

[31] S. Hosseinzadeh Kassani, P. Hosseinzadeh Kassani, M. J. Wesolowski, K. A. Schneider, and R. Deters, "Classification of Histopathological Biopsy Images Using Ensemble of Deep Learning Networks," in Proceedings of the 2019 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2019, pp. 1008-1013, doi: 10.1109/CSCI49370.2019.00188.