

A NOVEL AND EFFICIENT CBIR USING CNN FOR FLOWERS

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Engineering Degree in Electronics and Communication Engineering

By

SUBASH S I (Reg. No. **39130442**)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600 119**

APRIL - 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119
www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **SUBASH S I** (Reg. no. **39130442**) who carried out the project entitled "**A NOVEL AND EFFICIENT CBIR USING CNN FOR FLOWERS**", under my supervision from October 2022 to April 2023.

Mr.M.A.MUTHIAH, M.Tech.

Head of the Department

Dr. T. Ravi, M.E., Ph.D.

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I am **SUBASH S I** (Reg. No. **39130442**) hereby declare that the Project Report entitled “**A NOVEL AND EFFICIENT CBIR USING CNN FOR FLOWERS**”, done by me under the guidance of **Mr. M. A. MUTHIAH, M.Tech.**, Assistant Professor, Department of Electronics and Communication Engineering, at **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY, CHENNAI** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **ELECTRONICS AND COMMUNICATION**.

DATE:

PLACE: **Chennai**



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge our sincere thanks to the Board of Management of SATHYABAMA for their kind encouragement in doing this project and completing it successfully. I am grateful to them.

I convey my thanks to **Dr. N.M. NANDHITHA, M.E., Ph.D.**, Dean, School of Electrical and Electronics Engineering, and **Dr. T. RAVI, M.E., Ph.D.**, Head of the Department, Dept. of Electronics and Communication for providing me with the necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to our Project Guide **Mr. M. A. MUTHIAH, MTech.**, Assistant Professor, Department of Electronics and Communication Engineering, for his valuable guidance, suggestions, and constant encouragement paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication who were helpful in many ways for the completion of the project.

ABSTRACT

Image processing is vital to extract the required data from images. Machine learning is an efficient tool used for penetration in most of the classification and identification tasks performed by a computer. This project proposes the identification of a flower after the classification of flower images using a successful artificial intelligence tool named the Convolutional Neural Network (CNN). Models similar to this project have been used in most search engines for a long time, but CBIR (content-based image retrieval) still runs with less accuracy and produces outputs with fewer specifications due to the use of convolutional feed-forward networks for image retrieval. System performance depends a lot on the drawn-out features extracted from images. So, it is required to develop a CBIR system that retrieves similar images without explicit feature extraction and classification by using CNN, which accepts images as input. For experimentation, images from the Oxford-102 flower dataset are used.

CONTENTS

CHAPTER NO.	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF ABBREVIATIONS	ix
	LIST OF FIGURES	x
1.	INTRODUCTION	1
	1.1 Need for Flower Classification	2
	1.2 Evolution of Convolutional Neural Networks	4
	1.2.1 Simple and Complex Cells	4
	1.2.2 The Neocognitron	6
	1.2.3 CNN for Handwriting Recognition	6
	1.2.4 AlexNet	7
	1.3 Convolutional Neural Networks	7
	1.4 Working of a CNN	9
	1.5 History of Flower Classification	11
	1.6 Mathematical Description of Neural Networks	12
2.	LITERATURE SURVEY	16
3.	AIM AND SCOPE OF PRESENT INVESTIGATION	23
	3.1 Aim of the existing methodology	23
	3.2 Methodology used previously	23

3.2.1	Image Pre-processing	24
3.2.1.1	Image Pre-processing methods	24
3.2.1.2	Image feature extraction	25
3.3	Our idea	29
3.4	Resnet50	30
3.5	Resnet18	31
4.	EXPERIMENTAL METHODS USED	33
	PREVIOUSLY	
4.1	Existing Networks Used for Experimentation	33
4.1.1	Alexnet and its Specifications	33
4.1.1.1	Alexnet architecture	33
4.1.2	Googlenet	35
4.1.3	Mobilenetv2	36
4.1.4	Squeezenet	37
4.1.5	Shufflenet	38
4.1.6	Darknet19	39
4.1.7	Densenet121 model	40
4.2	Techniques can be used for increasing accuracy	41
4.3	Dataset	46
4.4	Experimental Results of existing model	47
5.	PROPOSED WORK	49
5.1	Residual Neural Network	50
5.2	Modified_Resnet-1	51

	5.3 Modified_Resnet-2	52
	5.4 Experimental Results	52
6.	CONCLUSION AND FUTURE SCOPE	56
	REFERENCES	57
	APPENDIX	60
	A. RESEARCH PAPER	60

LIST OF ABBREVIATION

CNN	Convolutional Neural Networks
CBIR	Content Based Image Retrieval
MNIST	Modified National Institute of Standards and Technology
ReLU	Rectified Linear Unit
SVM	Support Vector Machine
HSV	Hue, Saturation, Value
SURF	Speeded Up Robust Features
IEEE	Institute of Electrical and Electronics Engineers
VGG	Visual Geometry Group
SIFT	Scale Invariant Feature Transform
SFTA	Segmentation Based Fractal Texture Analysis
RF	Random Forest
KD	K-Dimensional
JPEG	Joint Photographic Experts Group
ADAM	Advanced Design And Modeling
RGB	Red Green Blue
CTM	Close To Metal

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
1.1	Inter-class similarity	3
1.2	Intra-class variability	3
1.3	Simple cell	5
1.4	Simple handwritten digits from the MNIST data set	7
1.5	Dimensions of an image	8
1.6	Considering a small patch from the actual image	8
1.7	Pooling	9
1.8	Feeding into locally connected network	10
1.9	Final Output (CNN)	11
1.10	Commonly used activation functions	13
3.1	Block diagram of classification work	23
3.2	The raw image and the target image aspect ratio	25
3.3	Mappings in Resnets	30
3.4	Architecture of resnet18	32
4.1	Alexnet Architecture	35
4.2	Mobilenetv2	37
4.3	Squeezenet Architecture	38
4.4	Shufflenet blocks	39
4.5	Architecture of densenet121 model	41
4.6	An illustration of the conventional models	42

4.7	Sample of vertically flip	42
4.8	ResNet architecture	43
4.9	Inceptionv3 architecture	44
4.10	Densenet architecture	44
4.11	Depth separable convolution	44
4.12	Mobilenet architecture	45
4.13	Image samples of 25 species from Oxford-102 Flower	46
4.14	Specificity of individual categories using Resnet50	48
4.15	Specificity of individual categories using Resnet18	48
4.16	Specificity of individual categories using Darknet19	48
5.1	Residual identity Block	50
5.2	Residual convolutional Block	51
5.3	Modified_Resnet-1	51
5.4	Modified_Resnet-2	52
5.5	Specificity of individual categories using Modified_Resnet-1	53
5.6	Specificity of individual categories using Modified_Resnet-2	53

CHAPTER 1

INTRODUCTION

The various deep learning methods use data to train neural network algorithms to do a variety of machine learning tasks, such as the classification of different classes of objects. Convolutional neural networks are deep learning algorithms that are very powerful for the analysis of images. This article will explain to you how to construct, train and evaluate convolutional neural networks. In the case of flower classification, image processing is a crucial step for computer-aided plant species identification. Flower image classification is based on the low-level features such as colour and texture to define and describe the image content. Colour features are extracted using normalized colour histogram and texture features are extracted using gray-level co-occurrence matrix. In this study, a dataset consists of 180 patterns with 7 attributes for each type of flower has been gathered. The finding from the study reveals that the number of images generated to represent each type of flower influences the classification accuracy. One interesting observation is that duplication of very hard to learn images assist Neural Network to improve its classification accuracy. This is also another area that could lead to better understanding towards the behaviour of images when applied to Neural Network classification.

Flowers are the most important producers of the earth that can grow in a wide variety of climates in terms of their habitats. They also keep on playing a very important role in the food chain by feeding almost all insect species in the world. In addition to this they play an important role in the food chain, and many drugs can be produced by using their healing properties. For such reasons, having a good knowledge of flowers and knowing their species is very important in terms of recognizing a new or rare plant species. Otherwise, many plants may be damaged because they are considered harmful to one's farmland or may be sold at very cheap prices. And all this occurs due to the inadequate recognition of the plant species. However, it is a real phenomenon that many of the plants grown in nature can be cultivated. In addition, increasing the recognition capacity of numerous endemic plant species, such as elecampane, verbascum thapsus whose life is limited to a specific area and grown only under special climatic conditions, will support the

development of the pharmaceutical industry. Thanks to the studies in this field, little-known plants will be able to see the true value they deserve.

Flowers can be seen everywhere in people's daily life, and flowers have great cultural value, economic value and ecological value in our life. Although there are significant differences in shape, structure and habits between flowers, there is a great deal of trouble in understanding and identifying flowers. Therefore, it is necessary to use a flower identification method to identify flowers quickly and correctly. With the rapid development of science and technology and the popularity of smart phones, people tend to use more and more vivid and easy-to-understand pictures instead of cumbersome words. However, the recognition rate of existing flower recognition experiments is relatively low, and the better methods are needed to support the more perfect identification of flowers. The classification of flower species has the following problems: Firstly, the traditional feature extraction mainly use the characteristics of color, shape and texture. The artificial selection of these features is more complicated and difficult. Secondly, the same flower at different times and in various environments showing different gestures, and the color and shape of flower species are very similar, which brings great difficulty to identification.

1.1 NEED FOR FLOWER CLASSIFICATION

There are about 369,000 named species of flowering plants in the world. Generally, experienced plant taxonomy experts can identify plants according to their flowers. However, it is difficult to distinguish these flowers for most people. To know the names or characteristics of the flowers, we usually consult with specialists, query with flower guidebooks or browse relevant web pages through keywords searching. An effective way to identify flower name can be done by classifying flower images, especially with the widely use of digital cameras, mobile phone, etc.

While the flower classification is appealing in its usefulness and meaningfulness, several restrictions have limited its realization. Unlike other obvious objects classification in which we need to distinguish obvious categories such as car and desk from each other, flower classification is a more difficult task because of inter-class similarity and a large intra-class variation. It is safe to say there are not any two same flowers in the world. What is more, it is tough to

distinguish the difference of some kinds that are similar in appearance even for people.

Figure. 1.1 shows examples of inter-class similarity, in which there are three different kinds of flowers but similar in appearance. Due to the non-rigid deformation of flowers, it is easy to be influenced by force during the process of blossom. In addition, images of flowers are often taken in a real environment where the illumination condition varies with the weather and time. Also, there is lot more variation in viewpoint, occlusions, scale of flower images. Figure. 1.2 shows intra-class variability because of illumination changes and view difference. All these problems lead to a confusion across classes and make the task of flower classification more challenging. In addition, the background also makes the flower classification task difficult. In this paper we address these limitations, providing techniques that are practical for flower classification problem.



Fig: 1.1: Inter-class similarity (Osteospermum, Pericallis, and Gazaniarigens)



Fig: 1.2: Intra-class variability (view difference)

There are enormous progress has been done in image based classification systems using large dataset. But flower classification is still complicated tasks due to large variety of flower classes have same shape, appearance or background information such as plant leaves and grass. So, flower classification is more challenging due to the difficulties of identifying flower classes. Furthermore, some flower classification algorithms give better classification accuracy. But, these classification algorithms depend on their features from a segmented flower regions only. Manual classification system takes huge amount of time and it is difficult to

manage with a large amount of flower images. An efficient flower classification system is very important in various applications such as plants monitoring systems, flower representation using content-based image retrieval and indexing, identifying live plants, floriculture industry and educational resources on flower taxonomy. Thus, robust techniques are important for classifying flower images. Thus, novel convenient method would be of great benefit for flower classification. Flower images show great diversity in color, shape and texture. Color, texture and shape information are used to differentiate one flower image class from other. So, Color, texture and shape information are widely used as features to identify the different flower classes. Color is used as one of the most important feature in flower image classification. Because it is more effective for individuals and machine vision. Color feature is also used to get more information about flowers for detecting and recognising flowers. To characterise the color feature of the flower images, color histogram of the flower region is used. Hence, texture is also another important characteristics which is used to classify and recognize objects and scenes. Petals, tepals or sepals are the main parts of the flowers. Crucial information about the flower classes are taken using these parts of the flowers. Also, some flowers have petals with very specific shape. So, shape is an important characteristics in flower classification. The aim of this paper is to concentrate on the flower classification technique that uses different combination of local features extracted from the flower images in order to improve classification performance.

1.2 EVOLUTION OF CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural networks, or CNNs for short, form the backbone of many modern computer vision systems. We will describe the origins of CNNs, starting from biological experiments of the 1950s below.

1.2.1 SIMPLE AND COMPLEX CELLS

In 1959, David Hubel and Torsten Wiesel described “simple cells” and “complex cells” in the human visual cortex. They proposed that both kinds of cells are used in pattern recognition. A “simple cell” responds to edges and bars of particular orientations, such as the fig 1.3

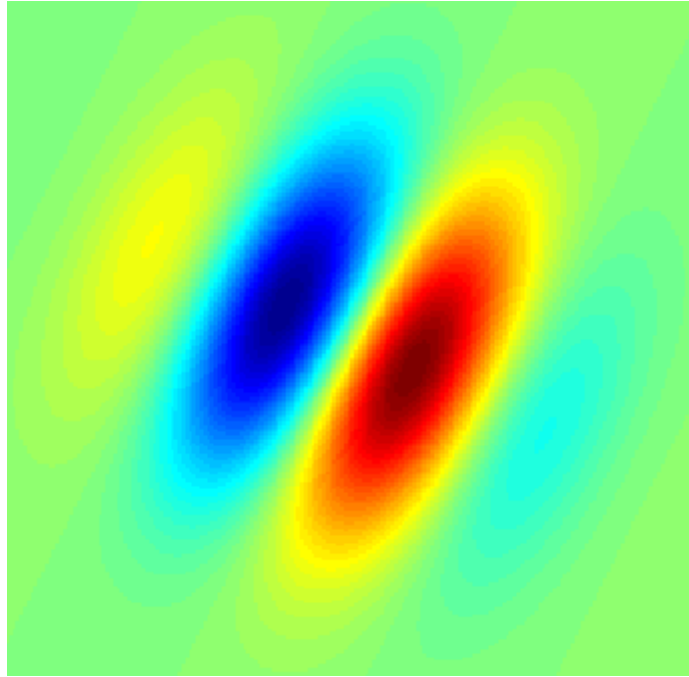


Fig: 1.3: Simple cell

A “complex cell” also responds to edges and bars of particular orientations, but it is different from a simple cell in that these edges and bars can be shifted around the scene and the cell will still respond. For instance, a simple cell may respond only to a horizontal bar at the bottom of an image, while a complex cell might respond to horizontal bars at the bottom, middle, or top of an image. This property of complex cells is termed “spatial invariance.”

Figure 1.3 diagrams the difference between simple and complex cells.

Hubel and Wiesel proposed in 1962 that complex cells achieve spatial invariance by “summing” the output of several simple cells that all prefer the same orientation (e.g. horizontal bars) but different receptive fields (e.g. bottom, middle, or top of an image). By collecting information from a bunch of simple cell minions, the complex cells can respond to horizontal bars that occur anywhere.

This concept – that simple detectors can be “summed” to create more complex detectors – is found throughout the human visual system, and is also the fundamental basis of convolution neural network models.

1.2.2 THE NEOCOGNITRON

In the 1980s, Dr. Kunihiko Fukushima was inspired by Hubel and Wiesel's work on simple and complex cells, and proposed the "neocognitron" model (original paper: "Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position"). The neocognitron model includes components termed "S-cells" and "C-cells." These are not biological cells, but rather mathematical operations. The "S-cells" sit in the first layer of the model, and are connected to "C-cells" which sit in the second layer of the model. The overall idea is to capture the "simple-to-complex" concept and turn it into a computational model for visual pattern recognition.

1.2.3 CONVOLUTIONAL NEURAL NETWORKS FOR HANDWRITING RECOGNITION

The first work on modern convolutional neural networks (CNNs) occurred in the 1990s, inspired by the neocognitron. Yann LeCun et al., in their paper "Gradient-Based Learning Applied to Document Recognition" demonstrated that a CNN model which aggregates simpler features into progressively more complicated features can be successfully used for handwritten character recognition.

Specifically, LeCun et al. trained a CNN using the MNIST database of handwritten digits (MNIST pronounced "EM-nisst"). MNIST is a now-famous data set that includes images of handwritten digits paired with their true label of 0, 1, 2, 3, 4, 5, 6, 7, 8, or 9. A CNN model is trained on MNIST by giving it an example image, asking it to predict what digit is shown in the image, and then updating the model's settings based on whether it predicted the digit identity correctly or not. State-of-the-art CNN models can today achieve near-perfect accuracy on MNIST digit classification.



Fig: 1.4: Sample handwritten digits from the MNIST data set.

One direct consequence of this work is that your mail is now sorted by machines, using automated handwriting recognition techniques to read the address.

1.2.4 ALEXNET

Throughout the 1990s and early 2000s, researchers carried out further work on the CNN model. Around 2012 CNNs enjoyed a huge surge in popularity (which continues today) after a CNN called AlexNet achieved state-of-the-art performance labeling pictures in the ImageNet challenge. Alex Krizhevsky et al. published the paper “ImageNet Classification with Deep Convolutional Neural Networks” describing the winning AlexNet model.

1.3 CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks are a special type of feed-forward artificial neural network in which the connectivity pattern between its neuron is inspired by the visual cortex.

The Convolutional Neural Networks, which are also called as covnets, are nothing but neural networks, sharing their parameters. Suppose that there is an image, which is embodied as a cuboid, such that it encompasses length, width, and

height. Here the dimensions of the image are represented by the Red, Green and Blue channels, as shown in the figure 1.5.

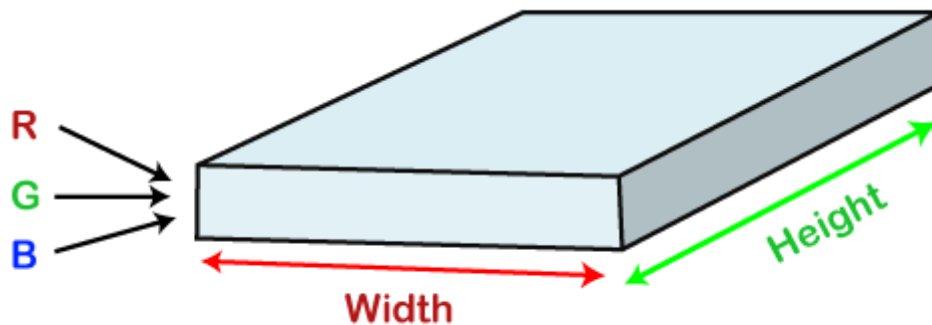


Fig:1.5: Dimensions of an image

Now assume that we have taken a small patch of the same image, followed by running a small neural network on it, having k number of outputs, which is represented in a vertical manner. Now when we slide our small neural network all over the image, it will result in another image constituting different width, height as well as depth. We will notice that rather than having R, G, B channels, we have come across some more channels that, too, with less width and height, which is actually the concept of Convolution. In case, if we accomplished in having similar patch size as that of the image, then it would have been a regular neural network. We have some weights due to this small patch.

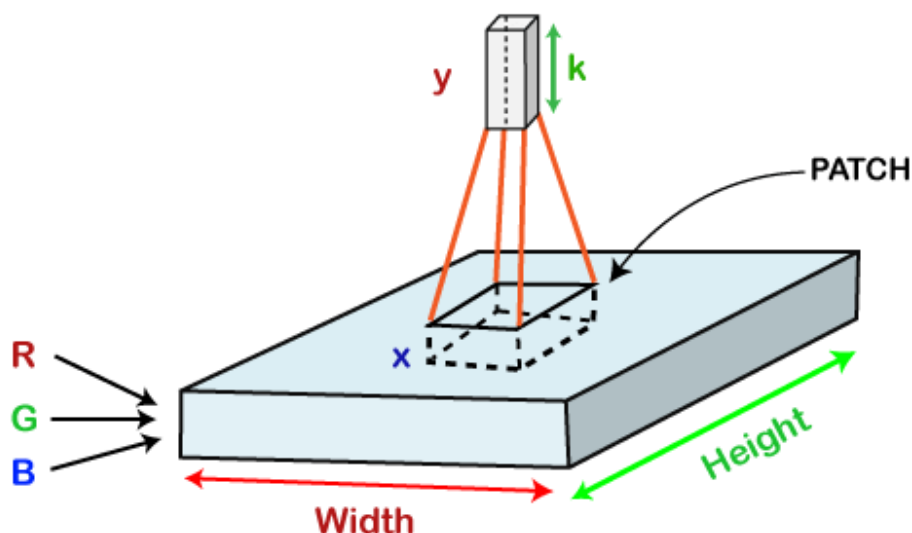


Fig: 1.6: Considering a small patch from the actual image

1.4 WORKING OF A CNN

Generally, a Convolutional Neural Network has three layers, which are as follows;

Input: If the image consists of 32 widths, 32 height encompassing three R, G, B channels, then it will hold the raw pixel([32x32x3]) values of an image.

Convolution: It computes the output of those neurons, which are associated with input's local regions, such that each neuron will calculate a dot product in between weights and a small region to which they are actually linked to in the input volume. For example, if we choose to incorporate 12 filters, then it will result in a volume of [32x32x12].

ReLU Layer: It is specially used to apply an activation function elementwise, like as $\max(0, x)$ thresholding at zero. It results in ([32x32x12]), which relates to an unchanged size of the volume.

Pooling: This layer is used to perform a down sampling operation along the spatial dimensions (width, height) that results in [16x16x12] volume.

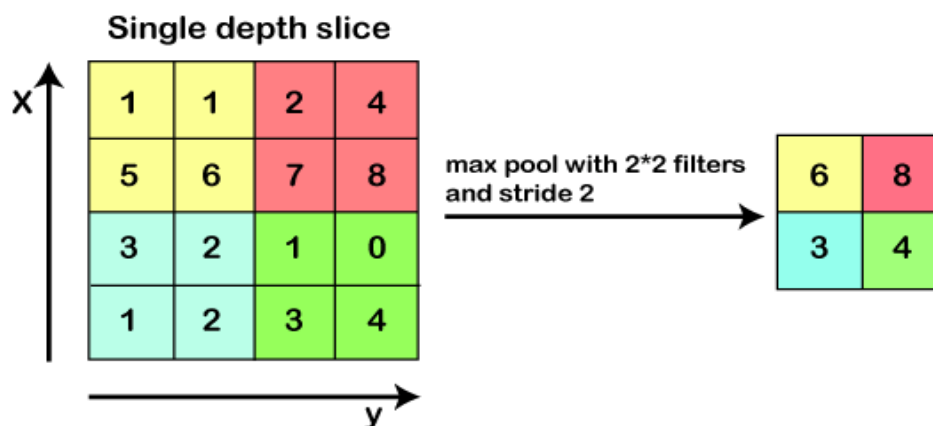


Fig: 1.7: Pooling

Locally Connected: It can be defined as a regular neural network layer that receives an input from the preceding layer followed by computing the class scores and results in a 1-Dimensional array that has the equal size to that of the number of classes

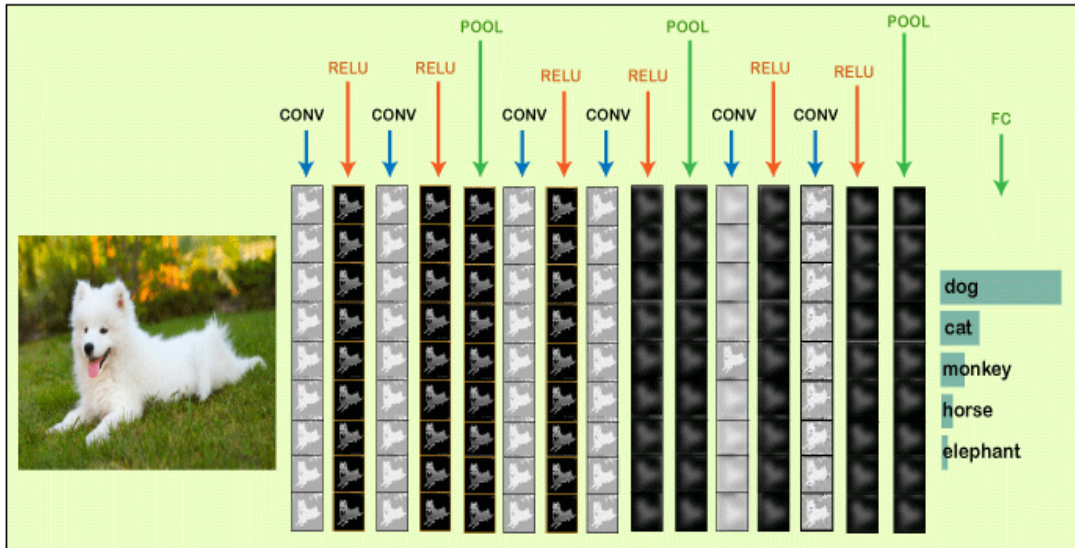


Fig: 1.8: Feeding into Locally Connected Network

We will start with an input image to which we will be applying multiple feature detectors, which are also called as filters to create the feature maps that comprises of a Convolution layer. Then on the top of that layer, we will be applying the ReLU or Rectified Linear Unit to remove any linearity or increase non-linearity in our images.

Next, we will apply a Pooling layer to our Convolutional layer, so that from every feature map we create a Pooled feature map as the main purpose of the pooling layer is to make sure that we have spatial invariance in our images. It also helps to reduce the size of our images as well as avoid any kind of overfitting of our data. After that, we will flatten all of our pooled images into one long vector or column of all of these values, followed by inputting these values into our artificial neural network. Lastly, we will feed it into the locally connected layer to achieve the final output.

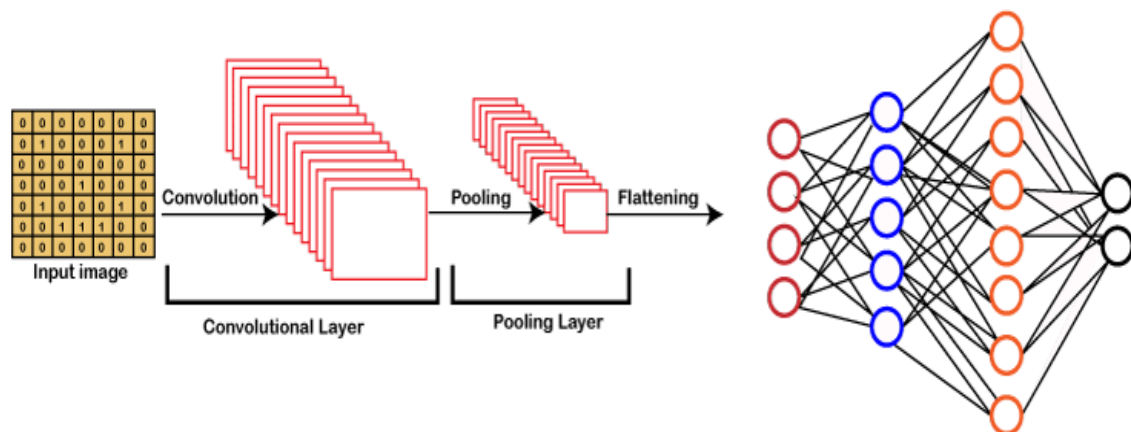


Fig: 1.9: Final Output (CNN)

1.5 HISTORY OF FLOWER CLASSIFICATION

Earlier flower recognition research was proposed by **Das et al.** However, only a color-based algorithm was developed. It is hard to classify different flower species based purely on color; many different flowers and species have similar colors, and many flowers of the same species have different colors.

Saitoh and Kaneto developed an automatic recognition system for wild flowers. There are two input images utilized; flower and leaf. The requirement for both flower and leaf features to be analysed is a restriction of this approach. In that system the piecewise linear discriminant function is used in the recognition process for 10 flower features and 11 leaf features. It is not convenient to take pictures of both flower and leaf. The system may require much time and space complexity.

Another approach by **Nilsback and Zisserman** extracted two main flower features, color and shape. Graph cut and color dependent conditional random forest algorithms were used to obtain color features. For shape features, generic shape fitting was used to detect petals. There is a problem of finding correct center of the flower. Some flowers have overlapping petals.

Chai developed a flower recognition process based on Nilsback and Zisserman's model. In the segmentation process used, the input image was divided into smaller sub-regions. Then, discriminative classifier and grabcut algorithms were applied. After that, a combination of feature selections, kernelization, SVM, heuristics and two unsupervised co-segmentation algorithms are used. It has a significant computational cost.

In 2014, flower recognition research via a mobile phone has been undertaken. For example, **Sunpetchniyon et al.** proposed an image search system for flowers on a mobile phone. This system used HSV histogram as a color feature and SURF as a shape feature. The accuracy of this proposed system has been poor.

Vuarnoz also developed a mobile application for flower recognition as a semester project. This project uses a water shed algorithm to extract the flower area. 18 different features encompassing colors and shapes of the flower are analysed. These features are mixed and matched to receive satisfied accuracy.

After these proposals several IEEE papers were published related to flower classification using CNN.

1.6 MATHEMATICAL DESCRIPTION OF NEURAL NETWORKS

It has been shown that neural networks are able to approximate any continuous function and thus have prevalence in the real world. Recent analysis has been done on the functions used in networks that work to classify images, approximate solutions to PDE's and ODE's, and generate languages. When trained with a proper amount of data, a neural network should be able to “learn” the function related to the task it is training on. The architecture of a classic feedforward network with L layers (depth) of n width is as follows:

$$\begin{aligned} z^{(l)} &= W^{(l)}a^{(l-1)} + b^{(l)}, \\ a^{(l)} &= \sigma(z^{(l)}), \\ u_{NN} &= W^{(L-1)}a^{(L)} + b^{(L-1)}, \end{aligned} \quad \text{-----equation 1.1}$$

where $a^{(0)}$ is the input to the network, u_{NN} is the output of the network and $\sigma: \mathbb{R} \rightarrow \mathbb{R}$ is the activation function. A traditional feedforward neural network begins with a set of inputs from a given piece of data; this data (depending on the original context) will be reshaped into a column vector where each element is a neuron in the input layer. In equation 1 the weights between each layer, W and biases, b are the trainable parameters of the network. Each layer is multiplied by the weights and a bias is added to compute a linear transformation, which is then put through an

activation function. Some commonly used examples of activation functions are ReLU, sigmoid, and hyperbolic tangent, which are seen in Figure 1.10.

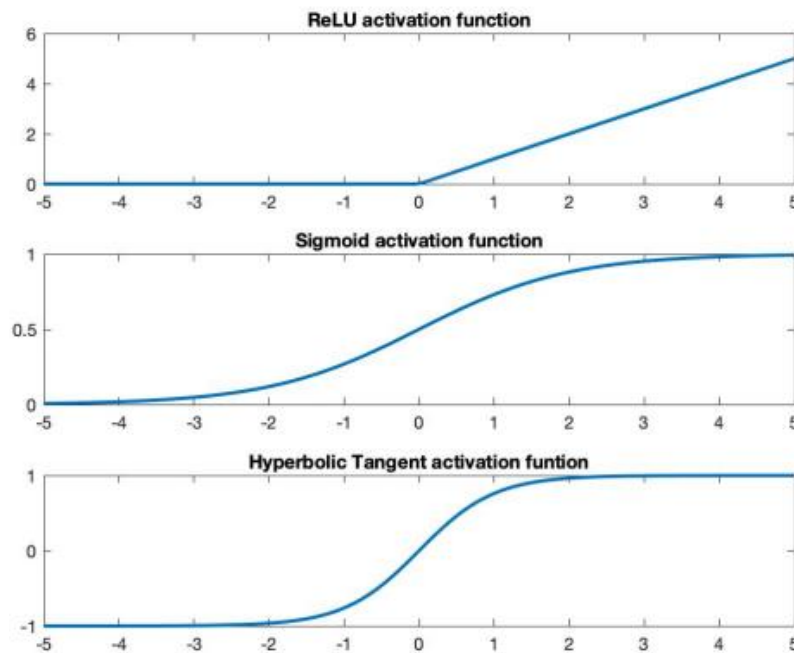


Figure 1.10: Commonly used activation functions

For the purposes of this paper, the ReLU activation function will be explored in more depth in later sections. It is worth noting that the functions in which neural networks attempt to model are often nonlinear, therefore, introducing non-linearity to the model is essential. This is done through the aforementioned activation functions such as ReLU and the sigmoid activation function. Overall, feedforward neural networks are composite functions that map from the input space, R_{in} , to the output space, R_{out} , and whose goal is to approximate a target function, for instance an underlying probability density or the solution to a differential equation. There are many different architectures of neural networks, some architectures are more appropriate for certain tasks; within the realm of feedforward networks, the key architectures are deep and shallow networks. Although there is not a consensus on what constitutes shallow or deep, a common understanding is that a network with 1 hidden layer is considered shallow, whereas a network with a large number of hidden layers is considered deep. Generally speaking, when using a ReLU activation function, deeper networks are able to approximate functions more effectively than very shallow networks with only one or two hidden layers because

the number of linear regions increases exponentially. This increase in linear regions can be thought of as an increase in the expressivity of the, or an improvement on its ability to approximate a desired unknown function. When working with neural networks, it must be trained (learn the function it is approximating), so there are two datasets - training and testing. In the case of classification problems, the training data will always have labels so that the performance of the network can be computed. The effectiveness of the network will be measured using what is referred to as a cost function. We denote the cost function by C_0 , which is a function of u_{NN} and therefore is a function of all network parameters, as well. There are many different types of cost functions, but each aims to compare each node of the output layer to the desired outcome and then pool the difference between the two as a measure of how correct or incorrect the network's output was. A non-exhaustive list of cost functions includes mean squared error, cross-entropy loss, and negative loglikelihood. The network's ability to "learn" and adjust the parameters to optimize the network is based on the cost function. While each network will have a unique goal, in general the training of the neural network is done to minimize the cost function so that it falls within a desired error tolerance (this tolerance will be highly context and application-dependent). In order to minimize the cost function, the standard technique is to perform stochastic gradient descent using backpropagation. Backpropagation is the process in which the network can adjust its weights and biases to minimize the cost function; it begins from the output layer and works its way to the first layer weights and biases. We will find the minimum of cost function by using the negative gradient since it is known that the gradient is the direction of the maximum of a given function, so the negative gradient will take us to the minimum. These gradients will be computed with respect to the weights and biases and are very cheap to compute.

$$\frac{\partial C_0}{\partial w^{(l)}} = \frac{\partial C_0}{\partial a^{(l)}} \cdot \frac{\partial a^{(l)}}{\partial z^{(l)}} \cdot \frac{\partial z^{(l)}}{\partial w^{(l)}} \quad \text{-----equation 1.2}$$

$$\frac{\partial C_0}{\partial b^{(l)}} = \frac{\partial C_0}{\partial a^{(l)}} \cdot \frac{\partial a^{(l)}}{\partial z^{(l)}} \cdot \frac{\partial z^{(l)}}{\partial b^{(l)}} \quad \text{-----equation 1.3}$$

Computing the gradient and negating it provides the direction to change the weights and biases for each epoch - a full run through the training data- which can be costly to run on a full set of data, depending on the size of the data set. To combat this costly calculation, a standard technique is to employ stochastic gradient descent which is a continuous, iterated nudging of the network in the opposite direction of the gradient computed over random batches of the training data. Once training is done over the specified number of epochs, we can compute both the training and testing accuracies from the model, where accuracy refers to the total number of correctly classified examples over the total number of examples tested. It is expected that the accuracy for the training data will be higher, but a significant difference between training and testing accuracy is an indication of overfitting - the network is overfit to the training data and no longer has capacity for variance in the test data- which can be a result of training for too many epochs.

CHAPTER 2

LITERATURE SURVEY

Faeze Sadati and Behrooz Rezaie presented a method for flower classification by using CNN for feature extractor. Unlike the traditional methods, CNN can extract features directly from input data. They have classified features obtained by a SVM machine learning method. The experiments have been implemented on two datasets. The datasets consist of natural images. By using SVM for classification, they have achieved 96.39% and 95.70% accuracy performance for the Oxford 17 flower and the Oxford 102 flower respectively. The experimental results show that the features obtained from CNN models with GAP is appropriate features for the classification and can achieve higher classification accuracy compared with the previous methods.

Muhammed YILDIRIM, Ahmet Çınar and Emine CENGİL have classified the images in the flowers image dataset. Their proposed model was more successful than both similar studies in the literature and the three pre-trained deep learning architectures used in their study. In addition, this study is of great importance for biologists. This method produces faster and more accurate results than traditional methods at that time. The study obtained an accuracy value of 94%. As can be understood from this value, this model can be used to classify flower types.

Rongxin Lv, Zhongzhi Li, Jiankai Zuo and Jing Liu presented a paper which constructs a flower classification model based on saliency detection and VGG-16 deep neural network model, uses stochastic gradient descent algorithm to update network weights, and uses Dropout method and transfer learning method to optimize the model. Experiments on the Oxford flower-102 data set show that the model has better classification results than traditional methods and other deep neural network architectures with a classification accuracy of 91.9%, which verifies the accuracy of this method for flower image classification tasks and the feasibility of flower identification.

Hossam M. Zawbaa, Mona Abbass, Sameh H. Basha et. al. proposed a system, it uses machine learning techniques to automatically identify flower category. The proposed approach includes three phases: segmentation, feature extraction, and classification phases. Segmentation phase aims to enhance the accuracy by

separating the flower shape from the image natural background. Then, flower features were extracted using two algorithms which are Scale Invariant Feature Transform (SIFT) and Segmentation-based Fractal Texture Analysis (SFTA). Classification phase can be run after the feature vectors are generated for each image. Support vector machine (SVM) and Random Forest (RF) classifiers are the used machine learning classification algorithms. the proposed approach has been done evaluated using 215 flower images for eight flowers categories. The experimental results show that the similarities on shape categories effects on the category classification accuracy. The categories that have shape different from other categories, achieve the height accuracy is achieved (100%) with different feature extractions and classifiers. As general, they declared that Support Vector Machine (SVM) classifier provides its better accuracy using the SIFT as a feature extraction algorithm. While SFTA algorithm gives better result with Random Forests (RF) classifier.

Shubhra Aich and Chil-Woo Lee presented a short paper, instead of using low number of original training images for flower classification, they had proposed an alternative with large number of generated artificial manifold images. Since, the between-class similarity is comparatively higher not only in flowers, but also in any kind of visually similar objects; Their proposal can be equally applicable in classifying other visually similar objects also.

Yong Wu,Xiao Qin , Yonghua Pan and Changan Yuan presented a model in which the feature extraction method of deep convolution neural network model used is more convenient and practical than the manual extraction feature method. At the same time, the neural network model added with the transfer learning method converges fast, and both robustness and generalization ability are stronger than random initialization model. Contrast with other traditional experimental methods to identify better. Experiments show that this method improves the accuracy of flower recognition and has a significant improvement.

Xudong Wei, Hongyu Zhang, Changkai Shi, Xinyue Yang, Huan Han, Bo Li presented a lightweight flower classification model based on MobileNetv3 small is constructed and the improved KD method is applied to the flower classification model. The experiment results show that the proposed method improves the accuracy of the lightweight flower classification model without increasing the

parameters and computation. The accuracy reaches 98.76% on Oxford102 flower, which has smaller size with very high accuracy compared with the traditional methods.

Dr.Shantala Giraddi, Dr.Shivanand Seeri, Dr.P.S.Hiremath and Dr.Jayalaxmi G.N presented a system in which the proposed algorithm is implemented using fine-tuned VGG16 on Google colab. The images are in JPEG format and in RGB color space. The Kaggle dataset is used in this study for training, validation and testing of fine-tuned VGG16 model. The dataset set contains five categories of flower images i.e. Daisy flower dataset has 813 flower images. Dandelion flower has a 1085 flower images, sunflower dataset has a 734 flower images, rose flower dataset has a 784 flower images, and tulip flower dataset has 984 flower images. The dataset consist of 4400 images and Images are about size 320x240 pixels. The dataset is divided into training dataset (80 %), validation dataset (10%) and testing dataset (10%). The small size 3x3 filters are used for the processing the images. The authors obtained highest accuracy of 95% with Fine-tuned VGG16 which is trained with flower dataset.

Yuanyuan Liu, Dengwen Zhou et. al. presented a model which extracts the features using the convolutional neural network, which are effective for a variety of object classification tasks. Experiments conducted on two different flower image databases consisting of 79 species and 102 species have shown that our proposed approach achieves higher classification accuracy than the previous methods. They collected a complex and challenging flower dataset to test our method, which contains 79 different species of flowers and 52,775 images.

Busra Rumeysa Mete and Tolga Ensari presented a system in which they obtained the highest result in the literature at that time with the proposed classification system and two flower datasets containing many classes (17 and 102). Thus, with the pre-trained model, they observed that the effect of both feature extraction and data augmentation on accuracy. In addition, we used t-sne model to visualize the images and how the features extracted from the images in the dataset are distributed. In addition, the classification system that we propose in this study can be used for different problems in image classification.

Sandip Desai, Chetan Gode and Punit Fulzele proposed a system regarding flower classification. For their experimentation they had used oxflower17 dataset. They had divided the dataset into two parts, 80% of images are used for training the neural network and remaining 20% images are used for validation. For training purpose, they had used ADAM as an optimizer and total 100 epochs are used. They got the following accuracy results during training and validation. From their observations, it is clear that as compared to the other state of art machine learning algorithms VGG-19 algorithm with transfer learning gives better result.

Neda Alipour, Mohammad Awrangjeb, Omid Tarkhaneh and Hongda Tian proposed a paper, based on the transfer learning approach and fine-tuning of a pre-trained model of DenseNet121, they trained a flower classifier on the Oxford-102 flower dataset. The results of classification showed that the proposed method could achieve higher accuracy than other methods at that time.

Tanakorn Tiay, Pipimphorn Benyaphaichit, and Panomkhawn Riyamongkol presented a system which is highly mobile, and can be used 'in the field' by researchers and laymen alike. In the proposed system, the original flower image is resized for faster processing. To obtain only flower in the image, the graph cut algorithm and RGB to grayscale conversion is used. After the system receives edge characteristics by Hu's seven-moment algorithm and color characteristics, K-nearest neighbor is used to classify flowers. Their system shows the most nearest 3 flower images with matching flower information of the first nearest flower image at that time. The accuracy of this system is more than 80%. Red rose and Butterfly pea, yield 100% accuracy because of their distinct color. This system can be further improved to yield more accuracy by combining other features, such as numbers of petals and flower texture.

Faisal Ridwan Siregar and Wikky Fawwaz Al Maki presented a flower classification algorithm. It employs a hybrid method of feature extraction, namely moment invariants as a shape feature and color moments as a color feature. They had Proved that we can develop a classification algorithm that yields accuracy of 88.74% using a dataset that has been augmented into 16368 images with various and large classes.

Pheeraphat Sarikabuta and Siriporn Supratid presented a paper. Their work investigates impacts of large -, medium - and small -size ResNets: ResNet50, ResNet35 and ResNet17 on Oxford-102 flower image dataset with distinct number of classes: 10, 50 and 102 flower classes. Within the scope of this study, ResNet50, ResNet35 and ResNet17 yield close accuracy results in very narrow ranges: [0.991, 0.998], [0.9975, 0.9996] and [0.9985, 0.9992] relying on 10, 50 and 102 classes of Oxford-102 flower image set, respectively. Even though ResNet35 produce a few bit better results than those two others for all cases, large increasing rate in classification performance is pointed by ResNet17 relative to ResNet50 and ResNet35, when expanding the flower dataset from 10 to 50 classes. Moreover, the increase in accuracy performance rate can be maintained based on such ResNet17, compared to ResNet50 and ResNet35 when enlarging the flower data from 50 to 102 classes. It can be noticed that even if ResNet35 yields the best results over others; less than 0.71% higher classification performance is indicated for ResNet35 than ResNet17 for all cases of 10-, 50- and 102-class flower data. Such a few bits insignificant superior performance, compared to approximate 33% higher amounts of parameters used in ResNet35 than ResNet17 may cause ResNet17 to be preferred to ResNet35.

Xiaoling Xia, Cui Xu and Bing Nan proposed a paper, based on the Inception-v3 model of TensorFlow platform, they used the transfer learning technology to train a flower classification model on flower category datasets. The classification accuracies of the model were 95% on Oxford-I7 flower dataset and 94% on Oxford-102 flower dataset, which is higher than other method at that time.

Fei Hu, Fuguang Yao and Changjiu Pu proposed a paper. In this paper, they had collected 1500 flower images from the internet and group them into six categories according to their types: muscari aucheri, bristlegrass, rose, cactus, sunflower and sakura. From each category, they had randomly selected 80% for training and the rest 20% for testing. The flower image classification has a good representation in the fine classification problem of images, and it also has some guiding significance for solving other fine classification problems.

Kishotha. S and B. Mayurathan proposed a flower classification method which was based on multiple feature descriptors. In that work, performance of SIFT, SURF, HSV, RGB and CTM features were analyzed in flower classification.

According to the experimental results, they had observed that multiple features empower the classifier to train a better model and achieve a better classification accuracy on the test sets. So, they strongly agreed to the use of combination of multiple local features in the flower image classification to get a better classification performance. In addition, the experimental results have shown that the combined (SURF + CTM) features outperform the individual features. The important thing to be noted in their work is that only color features with the combination of SIFT and SURF had given a good classification accuracy when compared to other results in their experiment.

Nitin R. Gavai, Yashashree A. Jakhade, Seema A. Tribhuvan and Rashmi Bhattad had experimented the flower classification with the Google's new model architecture, MobileNets found on depthwise distinguishable convolutional neural network. They proved how to build smaller and faster MobileNets application as flower classifier using width multiplier and resolution multiplier by trading off a rational amount of correctness to shrink size and latency. They concluded that, by validating MobileNet's efficiency when applied to an extensive range of image dataset. The MobileNets are optimized to become small and efficient by compromising on the accuracy aspect.

Emine CENGİL and Ahmet ÇINAR proposed a flower classification model, according to their study the importance of transfer learning approach is emphasized. Initialization of networks used for the classification task with pretrained networks rather than random weights is advantageous in terms of time and accuracy performance criteria. They had used five CNN models: AlexNet, Resnet18, GoogleNet, DenseNet201 and VGG16. According to the results of the five CNN models, multi-classification work showed acceptable accuracy performance for each model while highest accuracy was seen with VGG16 network.

Xiaojie Shi presented a system with an efficient feature extraction method used for image classification which was based on SPNs. The experiment results demonstrated that the new method could improve the classification precision.

Eisuke FUKUYAMA, Tomotaka KIMURA, Nobuhiko ITOH, and Takefumi HIRAGURI had performed an experiment, to discriminate pollination flowers during

artificial pollination by drone, learning was performed using images of various flowers. As a result, highly accurate identification was confirmed when the pollination flowers appeared large in the image. However, it was also confirmed that there were many problems, such as a lack of learning images and a change in brightness at the preprocessing stage.

CHAPTER 3

AIM AND SCOPE OF PRESENT INVESTIGATION

3.1 AIM OF THE EXISTING METHODOLOGY

The aim of the existing methodology is to build a flower image classification system that can retrieve and process the image to identify the flower species. It aims to be efficient, fast and lightweight.

3.2 METHODOLOGY USED PREVIOUSLY

The main tools used in the existing system are image processing and machine learning. The general conventional classification system goes as shown in figure 3.1

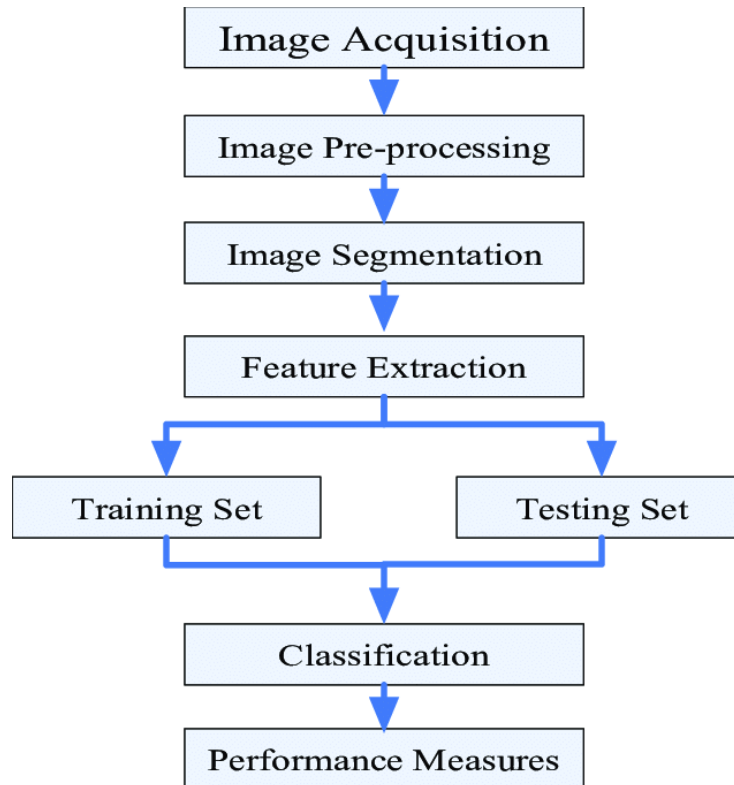


Fig: 3.1: Block diagram of Classification work

The conventional model uses a transfer learning approach for the process of flower classification. The construction process is divided into four fundamental components including,

- Image pre-processing
- Algorithm training
- Verification process
- Testing process

3.2.1 IMAGE PRE-PROCESSING

This stage is performed before feeding our images to the network, which can lead to better results from our method. As the input images were different in size and width, they resized all of them. After adjusting and normalizing the images, the model can be trained on them using their label. Then, those images will be fed to the fine-tuned model.

3.2.1.1 IMAGE PRE-PROCESSING METHOD

In the task of image processing, image preprocessing plays a very important role in the performance of the whole system. Most of current work in image preprocessing use general operations such as whitening and segmentation. Image classification algorithm on deep learning weakens image pre-processing steps, and it is desirable to use “raw data” as input as much as possible, but for the neural network model, it is necessary for that the input image size should be fixed. If we need the image size is $n \times n$, the original image needs to be scaled. If the high side is insufficient or exceeds n , it needs to stretch the raw image. Due to the requirement of the fixed size of the input image, the method is bound to sacrifice aspect ratio in some cases. Aspect ratio is the quotient of height divided by width or the quotient of width divided by height. The method of taking the raw image aspect ratio close to the target image aspect ratio of the image effect is relatively small, there is no loss of the original information even if zooming the image. However, for the other part of the raw image, aspect ratio is very different from the target, this method is not a good choice (see Figure 3.2)



Fig. 3.2: The raw image and the target image aspect ratio

The left image is the original, and the right is effect drawing after stretching. In order to preserve the true image information and get an accurate result during the experiment, we adopt an image preprocessing method based on aspect ratio preserving. Specifically, for the case where the aspect ratio of the raw image deviates from the target image, it is desirable to normalize to a fixed size. We choose to generate multiple copies of the image, each of which preserves the invariant aspect ratio. Although the method produces a redundant data image, it is best to keep the aspect ratio of the original image, and finally select the image with the highest degree of classification as the final result.

3.2.1.2 IMAGE FEATURE EXTRACTION

The important thing in the process of image classification is feature extraction. Because of the complexity of natural conditions, image feature extraction will encounter a great deal of troubles. It is usually desirable to select the features to achieve the following conditions:

- (1) feature vectors of belonging to the same class of images with similar expression.
- (2) feature vectors of different categories of images as much as possible divergence.
- (3) feature vector has good robustness to external disturbance factors.

For convenience, we consider $n \times n$ pixel images. As the different class of images may be similar in shape or color, the features of the whole image may not represent the characteristics of a certain class of flowers, the overall features of image is distinctly inaccurate. Therefore, the feature extraction of image is carried out by extracting the image patches. We adopt a novel feature extraction method (based on Coates and PCA as following.

(1) extract $m \times m$ pixels from all the train dataset, then deal with the zero mean and unit variance, in addition, ZCA whitening the patches.

(2) PCA algorithm is used on these small patches to learn a L dimensional feature.

(3) the $m \times m$ pixels extracted from each image of the validation set and test set, the processed are similar to step (1) and (2).

(4) each image in the dataset is given a $(n-m+1) \times (n-m+1) \times L$ dimensional feature vector. After pooling, a feature vector of $G \times G \times L$ dimension is obtained.

We adopt this feature extraction method based on three considerations. Extracting pixel patches could obtain image information from original image effectively and the ZCA with pooling are important step in deep learning, which could reduce dimension availably.

Some flowers have typical shapes, some are with specific colors, and some others have unique texture patterns, and some flowers are characterized by a combination of these properties. So different features are selected in order to describe different properties of a flower. Some of these feature descriptors that are used in this paper are described below.

Scale-invariant feature transform (**SIFT**): The scale-invariant feature transform (SIFT) is a well-known feature detection algorithm. SIFT is a technique for detecting salient and stable feature points in an image. For each and every such point, this algorithm also provides a set of features that characterize a small image region around the point. These features are invariant to rotation and scale. Unique features are extracted from gray-value images by filtering images at multiple scales and interest points which have sharp changes in local image intensities. Potential interest points which are invariant to scale and orientation are identified by using Difference of Gaussian (DoG) function over multiple scales. The local extrema of the DoG function is considered as interest points. 4×4 sized sub-region around the interest point is selected to computer the descriptor based on eight-orientation histograms, resulting in a 128-dimensional vector.

Speeded-Up robust features (**SURF**): SURF descriptor is a scale and rotation-invariant descriptor. Feature extraction and feature description are two main

steps in this algorithm. Fast-Hessian Detector based on very basic Hessian matrix approximation is used to identify the keypoints. Haar wavelets outputs are computed for orientation assignment. Wavelet outputs in a certain surrounding of the keypoint is used to form the keypoint descriptor. Eight orientation histograms at a 4 x 4 sub-region around the interest point is used to compute the extension to a length of 128 dimensional vectors.

HSV Color: Non-linear transform of RGB color is the HSV color. It describes perceptual color relationship and it gives more accuracy than RGB color space. Hue(H), saturation (S) and value (V) is used to compute HSV color. The property of color such as blue, green, and red is the Hue. Saturation is the intensity of the specific color. Value is calculated using brightness perception a specific color. HSV color is described by the HSV values of the pixels. HSV is the most common cylindrical-coordinate representations of points in an RGB color model. It is commonly used to generate high quality computer graphics and to select different colors needed for a particular image. HSV color model is defined by 3 values such as Hue, Saturation, Value. For each of these channels in an image, color moments are calculated. Therefore, an image is characterized by 9 moments: 3 moments for each 3 color channels.

RGB Histogram: RGB Histogram is the depiction of distribution of colors in image. RGB histogram for digital image is the representation of the number of pixels that have colors in each of a fixed list of color images. Histogram of red, green and blue luminance, are used to characterized each flower image.

Color Texture Moments (CTM): CTM are used to distinguish images based on their color feature. Color moments are calculated by using the mean, standard deviation and skewness of an image.

A preprocessing algorithm is employed in the flower classification system to improve image quality for inclusion in the feature extraction stage. In this connection, we perform manual image cropping, image segmentation, and image augmentation. Also, we change each image's color for each feature.

Manual Image Cropping: The flower images obtained from the dataset are very varied. Image cropping was performed to obtain the right image for processing. In this connection, image cropping reduced the image size to get a region of interest

(ROI). By performing image cropping, we could obtain representation of only one flower in each image.

Image Segmentation: The image segmentation process aims to separate the background image from the foreground image (object). In this stage, we employed the so-called GrabCut method to segment the image. The GrabCut method is proven to be one of the most sophisticated graph-based techniques. This method is a graph cut algorithm for segmenting color images repeatedly and efficiently.

In addition, the GrabCut method has many benefits for segmentation problems. GrabCut-based segmentation results are error-prone and sensitive to user interaction. The GrabCut method initializes the segmentation process as an interactive or semi-automated technique. GrabCut is the most appropriate method for binary label segmentation. It can segment an image into only the background and foreground areas. There have been several attempts conducted to modify GrabCut to be an automatic segmentation technique. The modification makes use of various unsupervised clustering techniques as an alternative to the GrabCut initialization phase. The results of segmentation of flower images using the GrabCut method.

Image Augmentation: We use the data augmentation process to obtain the better performance of the flower classification algorithm. In addition, it is employed to overcome image deficiencies in each class. This method has been widely used in image classification tasks. In this connection, we obtain new images by flipping the available images horizontally. Preprocessing for Shape Feature After performing augmentation on image data, each of image is converted into a grayscale image. Then, the grayscale image is converted to a binary image. The method proposed for shape features extraction does not require pixel values from the grayscale images. Since the dataset used in this research consists of flower images photographed from various positions, the well-known Otsu thresholding algorithm is needed to represent the object's position in the image. In, the Otsu method demonstrates satisfactory results in image segmentation. In our research, we employ the Otsu thresholding algorithm to the images segmented by the GrabCut algorithm. Finally, we employed the opening morphological transformation to the images. The opening morphological transformation is performed by applying erosion on the image. Then, the erosion result is eroded again. This process aims

to eliminate small objects around the flower image. The original images and their preprocessed results.

Preprocessing for Color Features: Each image input has a BGR color channel by congenital. Since the proposed color feature method for this feature uses HSV color channels, all image channels should be represented in HSV channels. Similar to the shape features, color features also employ the opening morphological transformation. The original sample image and preprocessed image.

3.3 OUR IDEA

To build an Efficient CBIR system using a suitable CNN to achieve the accuracy level at its best and would be higher than the proposed models.

The work is to modify the pre-trained networks, which would give better results than the actual ones.

These are the several improvements that can be made to the existing system. They are:

Accuracy -The accuracy of the present system can be improved.

Model loss - the loss of the system can be reduced.

Computation time - The computation time can be reduced and can be made.

Among the existing models, the highest level of accuracy is achieved in the resnet50. Resnet50 has a relatively large number of layers. So, in order to achieve better results with fewer layers, the modification is decided to be done in resnet18. Despite outperforming the other pre-trained networks in terms of accuracy, resnet50 has a higher layer count than resnet18. In most of the cases, for a huge dataset, the deeper networks will give best results, accordingly smaller networks will be suitable for small datasets. We chose to build a CNN with fewer layers, which can outperform large layered networks on both large and small datasets. As a result, the modification work is carried out in resnet18, which is less deep and also employs the feed-forward concept. The proposed system represents a network in which the actual resnet18, has some additional layers at the final layers, the additional layers were connected through series connections and no shortcuts were included.

3.4 RESNET50

ResNet50 is a variant of ResNet model which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. It has 3.8×10^9 Floating points operations. It is a widely used ResNet model and we have explored **ResNet50 architecture** in depth. The ResNets were initially applied to the image recognition task but as it is mentioned in the paper the framework can also be used for non-computer vision tasks also to achieve better accuracy.

As we know Deep Convolutional neural networks are really great at identifying low, mid and high-level features from the images, and stacking more layers generally gives us better accuracy so a question arises that is getting better model performance as easy as stacking more layers? With this question arises the problem of vanishing/exploding gradients those problems were largely handled in many ways and enabled networks with tens of layers to converge but when deep neural networks start to converge, we see another problem of the accuracy getting saturated and then degrading rapidly and this was not caused by overfitting as one may guess and adding more layers to a suitable deep model just increased the training error. This problem was further rectified by taking a shallower model and a deep model that was constructed with the layers from the shallow model and adding identity layers to it and accordingly, the deeper model shouldn't have produced any higher training error than its counterpart as the added layers were just the identity layers.

The authors addressed this problem by introducing deep residual learning framework so for this they introduce shortcut connections that simply perform identity mappings.

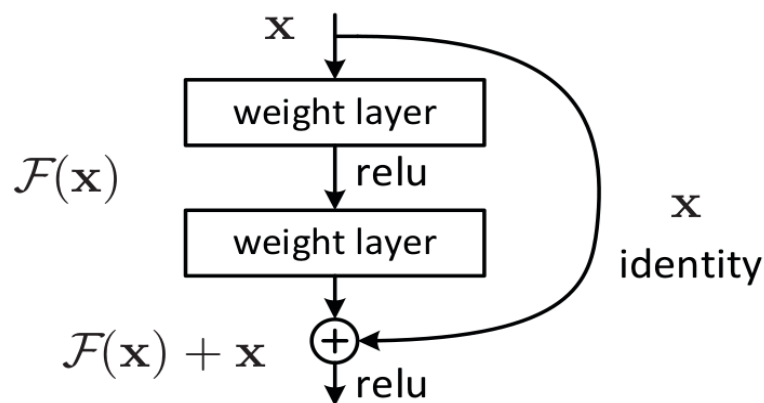


Fig 3.3: Mappings in Resnets

To demonstrate how much better the ResNet are they compared it with a 34-layer model and an 18 layer model both with plain and residual mappings and the results were not so astounding the 18 layer plain net outperformed the 34 layer plain net and in the case of ResNet the 34 layer ResNet outperformed the 18 layer ResNet.

There was a small change that was made for the ResNet 50 and above that before this, the shortcut connections skipped two layers but now, they skip three layers and also there were 1×1 convolution layers added that we are going to see in detail with the ResNet 50 Architecture.

Convolution with a kernel size of 7×7 and 64 different kernels all with a stride of size 2 giving us 1 layer. Next, we see max pooling with also a stride size of 2. In the next convolution, there is a $1 \times 1, 64$ kernel following this a $3 \times 3, 64$ kernel, and at last a $1 \times 1, 256$ kernel, these three layers are repeated in total 3 times so giving us 9 layers in this step. Next, we see the kernel of $1 \times 1, 128$ after that a kernel of $3 \times 3, 128$, and at last a kernel of $1 \times 1, 512$, this step was repeated 4 times so giving us 12 layers in this step. After that, there is a kernel of $1 \times 1, 256$ and two more kernels with $3 \times 3, 256$ and $1 \times 1, 1024$ and this is repeated 6 times giving us a total of 18 layers. And then again, a $1 \times 1, 512$ kernel with two more of $3 \times 3, 512$, and $1 \times 1, 2048$, and this was repeated 3 times giving us a total of 9 layers. After that, we do an average pool and end it with a fully connected layer containing 1000 nodes and at the end a SoftMax function so this gives us 1 layer. We don't actually count the activation functions and the max/ average pooling layers. So, totaling this it gives us a $1 + 9 + 12 + 18 + 9 + 1 = 50$ layers Deep Convolutional network.

3.5 RESNET18

ResNet18 is a 72-layer architecture with 18 deep layers. The architecture of this network is aimed at enabling large amounts of convolutional layers to function efficiently. However, the addition of multiple deep layers to a network often results in a degradation of the output. This is known as the problem of vanishing gradient where neural networks, while getting trained through back propagation, rely on the gradient descent, descending the loss function to find the minimizing weights. Due to the presence of multiple layers, the repeated multiplication results in the gradient becoming smaller and smaller thereby “vanishing” leading to a saturation in the network performance or even degrading the performance.

The primary idea of ResNet is the use of jumping connections that are mostly referred to as shortcut connections or identity connections. These connections primarily function by hopping over one or multiple layers forming shortcuts between these layers. The aim of introducing these shortcut connections was to resolve the predominant issue of vanishing gradient faced by deep networks. These shortcut connections remove the vanishing gradient issue by again using the activations of the previous layer. These identity mappings initially do not do anything much except skip the connections, resulting in the use of previous layer activations. This process of skipping the connection compresses the network; hence, the network learns faster. This compression of the connections is followed by expansion of the layers so that the residual part of the network could also train and explore more feature space. The input size to the network is $224 \times 224 \times 3$, which is predefined. The network is considered to be a DAG network due to its complex layered architecture and because the layers have input from multiple layers and give output to multiple layers. Residual networks and their variants have broadly been implemented for the analysis of medical images. Figure 3.4 shows the layer architecture of ResNet18 CNN model.

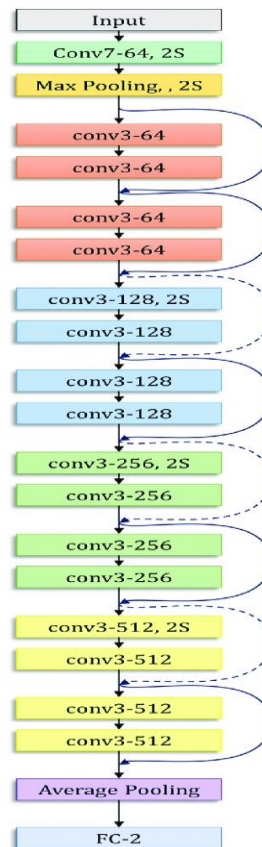


Fig 3.4: Architecture of Resnet18

CHAPTER 4

EXPERIMENTAL METHODS USED PREVIOUSLY

4.1 EXISTING NETWORKS USED FOR EXPERIMENTATION

The pre-trained networks used for experimentation are Alexnet, Googlenet, Mobilenetv2, Nasnetmobile, Squeezenet, Resnet18, Resnet50, Shufflenet, and Darknet19.

4.1.1 ALEXNET AND ITS SPECIFICATIONS

Alexnet won the Imagenet large-scale visual recognition challenge in 2012. The model was proposed in 2012 in the research paper named Imagenet Classification with Deep Convolution Neural Network by Alex Krizhevsky and his colleagues. In this model, the depth of the network was increased in comparison to Lenet-5.

The Alexnet has eight layers with learnable parameters. The model consists of five layers with a combination of max pooling followed by 3 fully connected layers and they use Relu activation in each of these layers except the output layer.

They found out that using the relu as an activation function accelerated the speed of the training process by almost six times. They also used the dropout layers, that prevented their model from overfitting. Further, the model is trained on the Imagenet dataset. The Imagenet dataset has almost 14 million images across a thousand classes.

4.1.1.1 ALEXNET ARCHITECTURE

One thing to note here, since Alexnet is a deep architecture, the authors introduced padding to prevent the size of the feature maps from reducing drastically. The input to this model is the images of size $227 \times 227 \times 3$. The first convolution layer with 96 filters of size 11×11 with stride 4. The activation function used in this layer is relu. The output feature map is $55 \times 55 \times 96$. In case, you are unaware of how to calculate the output size of a convolution layer use, "Output= ((Input-filter)/stride)+1".

Also, the number of filters becomes the channel in the output feature map. Next, we have the first Maxpooling layer, of size 3×3 and stride 2. Then we get the resulting feature map with the size $27 \times 27 \times 96$.

After this, we apply the second convolution operation. This time the filter size is reduced to 5X5 and we have 256 such filters. The stride is 1 and padding 2. The activation function used is again relu. Now the output size we get is 27X27X256.

Again, we applied a max-pooling layer of size 3X3 with stride 2. The resulting feature map is of shape 13X13X256. Now we apply the third convolution operation with 384 filters of size 3X3 stride 1 and also padding 1. Again, the activation function used is relu. The output feature map is of shape 13X13X384.

Then we have the fourth convolution operation with 384 filters of size 3X3. The stride along with the padding is 1. On top of that activation function used is relu. Now the output size remains unchanged i.e 13X13X384.

After this, we have the final convolution layer of size 3X3 with 256 such filters. The stride and padding are set to one also the activation function is relu. The resulting feature map is of shape 13X13X256. So if you look at the architecture till now, the number of filters is increasing as we are going deeper. Hence it is extracting more features as we move deeper into the architecture. Also, the filter size is reducing, which means the initial filter was larger and as we go ahead the filter size is decreasing, resulting in a decrease in the feature map shape. Next, we apply the third max-pooling layer of size 3X3 and stride 2. Resulting in the feature map of the shape 6X6X256. After this, we have our first dropout layer. The drop-out rate is set to be 0.5. Then we have the first fully connected layer with a relu activation function. The size of the output is 4096. Next comes another dropout layer with the dropout rate fixed at 0.5.

This followed by a second fully connected layer with 4096 neurons and relu activation. Finally, we have the last fully connected layer or output layer with 1000 neurons as we have 10000 classes in the data set. The activation function used at this layer is Softmax.

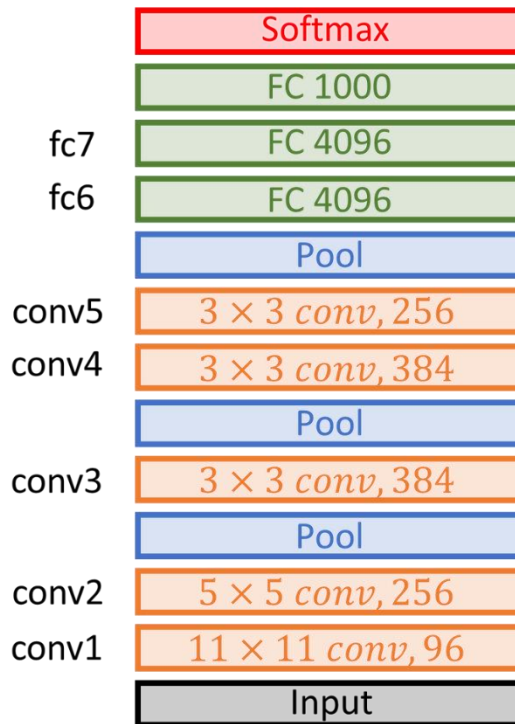


Fig 4.1: Alexnet Architecture

This is the architecture of the Alexnet model. It has a total of 62.3 million learnable parameters.

4.1.2 GOOGLNET

GoogLeNet, which is famously known as Inception Net, is a Deep Learning model built by researchers at Google. Going Deeper with Convolutions was the paper by which the GoogLeNet Model first came into existence.

The ILSVRC (ImageNet Large Scale Visual Recognition Challenge) evaluates models based on their object detection and image classification performance every year. GoogLeNet was declared the winner in ILSVRC 2014 for the image classification challenge. The performance of GoogLeNet was significantly better than VGG Model (runner-up). Also, the error rate was reduced to a much greater extent when compared with 2013 and 2012 ILSVRC winners ZF-Net and AlexNet, respectively. GoogleNet introduced the inception module in deep learning.

There are 22 Parameterized Layers in the Google Net architecture; these are Convolutional Layers and Fully-Connected Layers; if we include the non-parameterized layers like Max-Pooling, there are a total of 27 layers in the GoogLeNet Model.

Input - The GoogLeNet model takes an input image of 224×224 .

Output - The output layer (or the softmax layer) has 1000 nodes that correspond to 1000 different classes of objects.

Inception Module - The idea of the Inception module is to bring down the number of parameters in a Deep Neural Network; the inception module is built with many small convolutions. The models like AlexNet have 60 million parameters, whereas GoogleNet had only 4 million parameters also; the architecture of GoogleNet was much deeper than AlexNet. GoogleNet model has nine identical units known as inception units or inception modules.

There are Convolutional Kernels of different shapes 1x1, 3x3, & 5x5 in the Inception Module. The output of all these kernels is stacked together at the end of the inception unit. Larger convolutional kernels cover a large area in an image to get the information, and the smaller kernels work on a smaller area in the image. So, a 1x1 convolutional kernel will give us a finer detail in the image when compared with a 5x5 convolutional kernel.

Auxiliary Classifier - Auxiliary Classifier is a classification unit added twice in the middle of the GoogleNet network. We use Auxiliary Classifiers to tackle the problem of vanishing gradient. Each auxiliary classifier has a 5x5 average pooling layer, a 1x1 convolutional layer, two fully connected layers with 1024 units, and a softmax layer with 1000 units. The auxiliary classifier units are added on top of the middle inception units. When we train the neural network, the loss of the auxiliary classifier units is multiplied by 0.3 and then added to the total network loss.

4.1.3 MOBILENETV2

In MobileNetV2, a better module is introduced with **an** inverted residual structure. Non-linearities in narrow layers are removed this time. With MobileNetV2 as the backbone for feature extraction, state-of-the-art performances are also achieved for object detection and semantic segmentation.

In MobileNetV2, there are two types of blocks. One is residual block with stride of 1. Another one is block with stride of 2 for downsizing. There are 3 layers for both types of blocks. This time, the first layer **is** 1x1 convolution with ReLU6. The **second layer** is the depthwise convolution. The third layer is another 1x1 convolution but without any non-linearity. It is claimed that if ReLU is used again, the deep networks only have the power of a linear classifier on the non-zero volume part of the output

domain. And there is an expansion factor t . And $t=6$ for all main experiments. If the input got 64 channels, the internal output would get $64 \times t = 64 \times 6 = 384$ channels.

where t : expansion factor, c : number of output channels, n : repeating number, s : stride. 3×3 kernels are used for spatial convolution. In typical, the primary network (width multiplier 1, 224×224), has a computational cost of 300 million multiply-adds and uses 3.4 million parameters. The performance trade offs are further explored, for input resolutions from 96 to 224, and width multipliers of 0.35 to 1.4. The network computational cost up to 585M MAdds, while the model size varies between 1.7M and 6.9M parameters. To train the network, 16 GPU is used with batch size of 96.

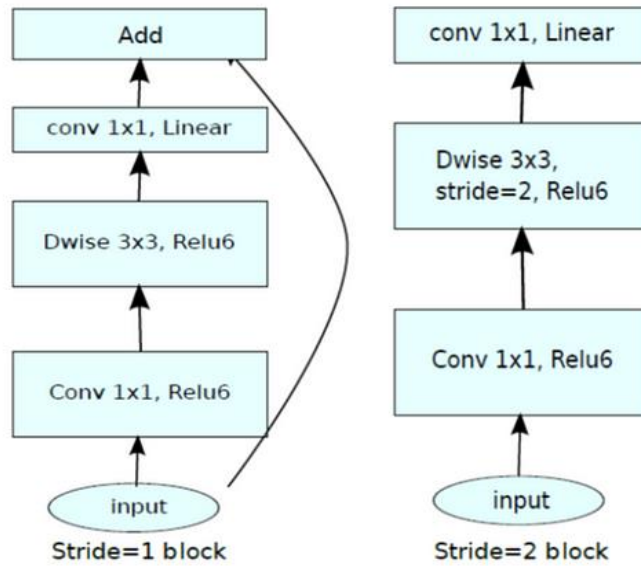


Fig 4.2: Mobilenetv2 Architecture

4.1.4 SQUEEZENET

The SqueezeNet architecture begins with a standalone convolution layer (conv1), followed by 8 Fire modules (fire2-9), ending with a final conv layer (conv10). The number of filters per fire module is gradually increased from the beginning to the end of the network. SqueezeNet performs max-pooling with a stride of 2 after layers conv1, fire4, fire8, and conv10; these relatively late placements of pooling are per Strategy 3. The entire macroarchitectural view of the SqueezeNet architecture is showcased below. The leftmost shows SqueezeNet, the middle one is SqueezeNet with simple bypass, and the rightmost one is SqueezeNet with complex bypass.

1-pixel border of zero-padding is added in the input data to 3x3 filters of expand modules. This makes the output activations from 1x1 and 3x3 filters have the same height and width. ReLU (rectified linear unit) is the activation function used in the squeeze and expand layers. Dropout regularization with a ratio of 50% is added after the fire9 module. The lack of fully connected layers is inspired by the NiN architecture. The learning rate while training SqueezeNet starts with 0.04, and is linearly decreased throughout the training. Refer to [this](#) link for details on the training parameters. Below is a figure that comprises of the SqueezeNet architectural dimensions.

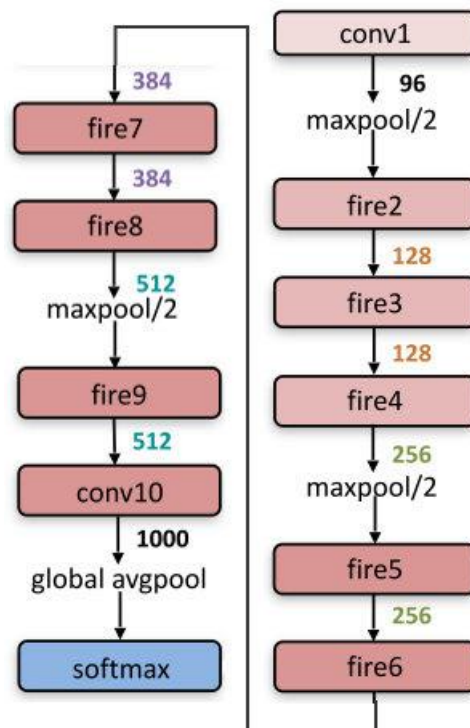


Fig 4.3 Squeezenet Architecture

4.1.5 SHUFFLENET

It is composed of a stack of ShuffleNet units grouped into three stages. In Table 1, the group number **g** controls the connection sparsity of pointwise convolutions. At the same time, **g** is assigned to be different numbers, so the output channels can be computed and evaluated to ensure the total computational costs are approximately the same (~140 MFLOPs).

The network can be freely customized to a desired complexity. To accomplish this, simply apply a scale factor **s** on the number of channels. For example, if the

networks are denoted in Table 1 as “ShuffleNet 1×”, then “ShuffleNet s×” means scaling the number of filters in ShuffleNet 1× by s times, thus overall complexity will be roughly s squared times of ShuffleNet 1×.

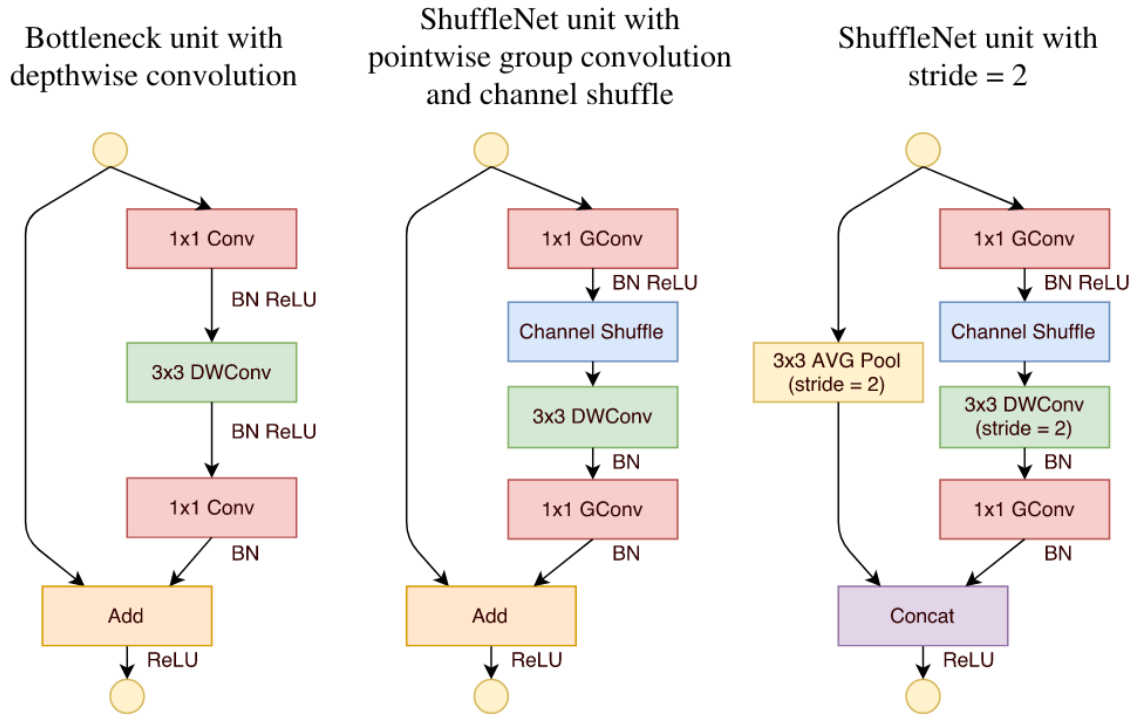


Fig 4.4: Shufflenet Blocks

4.1.6 DARKNET19

The Darknet19 classification model is to be used as the base of YOLOv2. The model builds off of prior work on network design as well as common knowledge in the field. Similar to the VGG models we use mostly 3×3 filters and double the number of channels after every pooling step. Following the work on Network in Network (NIN) we use global average pooling to make predictions as well as 1×1 filters to compress the feature representation between 3×3 convolutions. We use batch normalization to stabilize training, speed up convergence, and regularize the model. Our final model, called Darknet-19, has 19 convolutional layers and 5 max-pooling layers.

Training for classification -The network is trained on the standard ImageNet 1000 class classification dataset for 160 epochs using stochastic gradient descent with a starting learning rate of 0.1, polynomial rate decay with a power of 4, weight decay of 0.0005 and momentum of 0.9 using the Darknet neural network

framework. During training, we use standard data augmentation tricks including random crops, rotations, hue, saturation, and exposure shifts. As discussed above, after our initial training on images at 224×224 we fine tune our network at a larger size, 448. For this fine-tuning we train with the above parameters but for only 10 epochs and starting at a learning rate of 10^{-3} .

Training for detection - The network is modified by removing the last convolutional layer and instead adding on three 3×3 convolutional layers with 1024 filters each followed by a final 1×1 convolutional layer with the number of outputs we need for detection. For VOC we predict 5 boxes with 5 coordinates each and 20 classes per box so 125 filters. Also, a passthrough layer from the final $3 \times 3 \times 512$ layer is added to the second to last convolutional layer so that the model can use fine-grain features.

4.1.7 DENSENET121 MODEL

In this stage, they had focused on the transfer learning process by fine-tuning their pre-trained model. Transfer learning means they do not train the models from scratch because it is time-consuming and requires a considerable amount of data for training. Thus, they just transferred the features using employing a pre-trained model for their task, which was trained before on the ImageNet dataset, which was a dataset consisting of millions of mages in 1000 classes. So, in the fine-tuning procedure, they replaced the classification function to compute the probability of their 102 classes. With the help of Pytorch framework they retrained the DenseNet's final layer for the 102 classes by using fine-tuning. They kept the parameters of the previous layers fixed and removed the classifier layer of their model, and then retrained it again. The pre-trained model they employed for their method was DeseNet121. DenseNet121 is a convolutional neural network with 24 layers where each layer is connected to all other layers that are deeper in the network. In other words, each block is an iterative concatenation of previous layers. The output of each layer was calculated using the equation (4.1)

$$x_l = H_l([x_{l-1}, x_{l-2}, \dots, x_0]) \quad \text{----- equation 4.1}$$

Where x_l is the output of the l th layer, H_l is a non-linear transformation applied to the output of the previous layers of $[x_{l-1}, x_{l-2}, \dots, x_0]$. DenseNet has fewer parameters than an equivalent traditional CNN, as there is no need to learn

redundant feature maps. On the other hand, DenseNet layers are very narrow (e.g., 12 filters), and they just add a small set of new feature-maps. Also, each layer of DenseNet has direct access to the gradients from the loss function and the original input image; therefore, there are not any problems related to the flow of information and gradients during training procedure. The structure of DenseNet model is shown in Fig.4.5

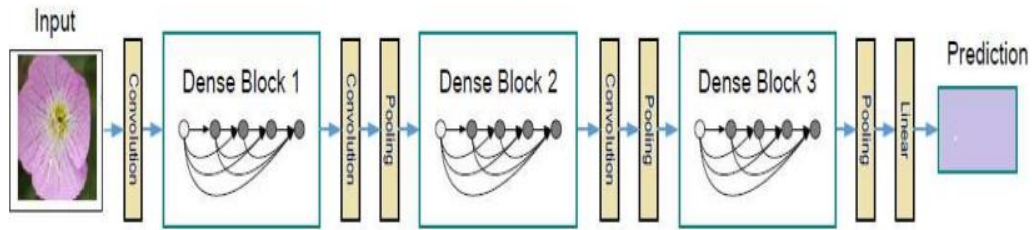


Fig: 4.5: Architecture of densenet121 model

4.2 TECHNIQUES CAN BE USED FOR INCREASING ACCURACY

Most of the conventional methods used some techniques in three stages:

- A. Data augmentation
- B. Feature extraction
- C. Classification First

They increased the data by the aid of data augmentation techniques. Then, they had used them as the inputs for pre-trained CNN models, i.e., ResNet101, InceptionV3, DenseNet121, and MobileNet. They had also utilized the GAP layer instead of the classification part for the purpose of deep feature extraction. The extracted feature vectors were concatenated to form a single deep feature. In the last stage, they classified single deep feature with SVM classifier for the determination of the labels of the flower images. These stages are shown in Fig. 4.6.

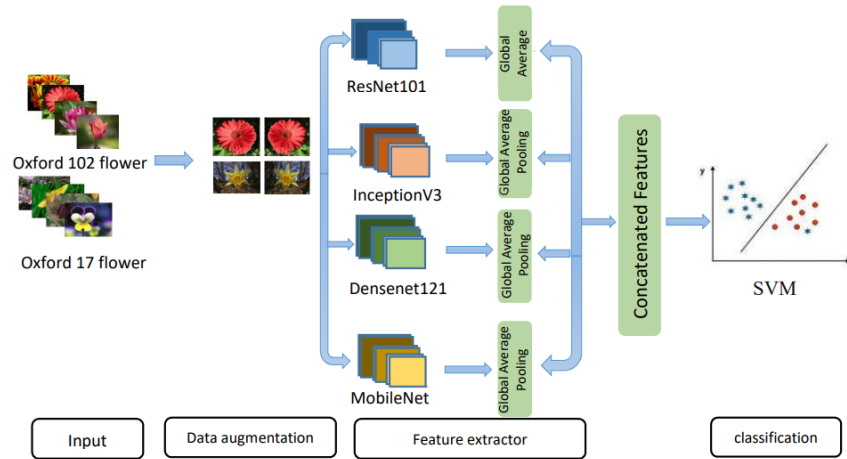


Fig: 4.6: An illustration of the conventional models

A. Data Augmentation

In Data Augmentation, they had expanded the datasets with one of the data augmentation techniques. In order to obtain better performance on the image classification and thereby to fix dataset shortage. In this paper, they had just doubled the dataset using vertical flip in both datasets as shown in Fig.4.7.



Fig:4.7: Sample of vertically flip

B. Feature Extraction

A simple model of CNN was firstly proposed in 1989. Nowadays, the CNN models have been developed and there are several CNN models available. CNN models include several layers. The first layers extract low-level features like edge information. The middle layers extract intermediate-level features like simple geometric shapes and the last layers extract high-level features or actually more specific features. In this study, they had used four CNN models, i.e., ResNet101,

InceptionV3, DenseNet121, and MobileNet, those were previously trained on ImageNet competition. The models have successful results on ImageNet challenges. As these models are used as feature extractors, we can remove the classification part and add a GAP layer instead of it.

The main reasons for using GAP layer are:

- 1) Picking up the presence of features in an image with the averaging operation.
- 2) Reducing the competitions cost.

Now, I briefly explain these models. The ResNet was introduced to solve the problems of increasing network depth. The main idea of the ResNet model is in utilizing Residual block. In this network, communication is considered outside the convolutional structure to move the input of the previous layer to the next layer without intermediaries. These connections are called Skip connections and this structure is called a Residual block. The architecture of this model is shown in Fig.4.8.

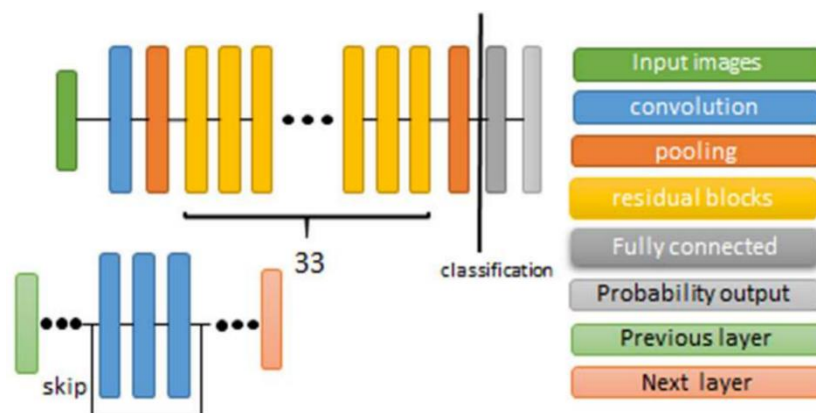


Fig: 4.8: ResNet architecture

InceptionV3 model is one of the GoogleNet models. This model unlike the other models, it is not sequential. The layers are placed in parallel, and each of these blocks is called an Inception module. This idea caused extracted features in the different levels. The architecture of this model is given in Fig.4.9.

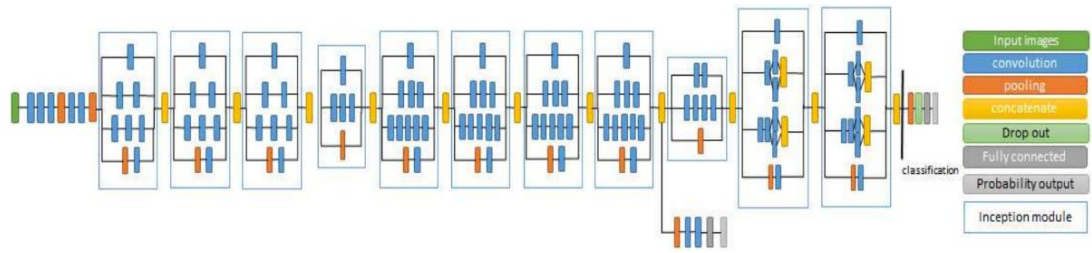


Fig: 4.9: InceptionV3 architecture

DenseNet is a CNN model in which each layer is connected to all other layers that are deeper in the network, to enable maximum information flow between the layers of the network. To preserve the feed-forward nature, each layer obtains inputs from all the previous layers and passes on its own feature maps to all the layer come after it. The architecture of this model is illustrated in Fig.4.10.

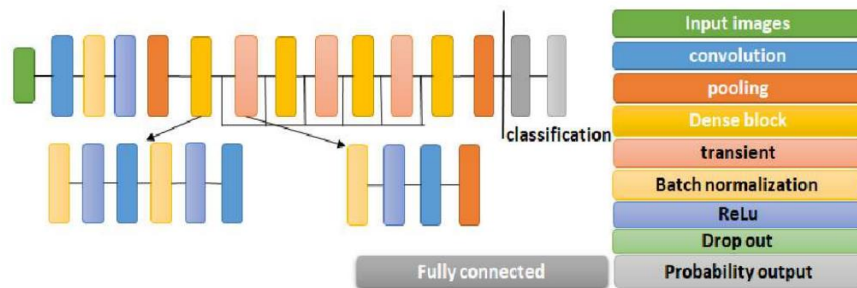


Fig: 4.10: DenseNet architecture

MobileNet uses depth separable convolutions as shown in Fig.4.11. Depth separable convolution consists of a depth-wise convolution that is the channel-wise $D_k \times D_k$ spatial convolution and a point-wise convolution that is the 1×1 convolution. It significantly reduces the number of the parameter when compared to the network with regular convolutions with the same depth in the nets.

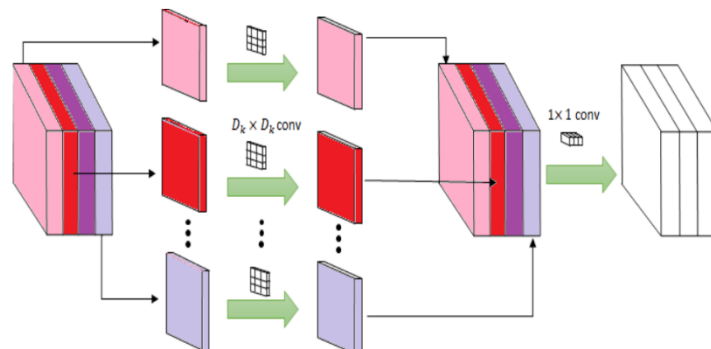


Fig: 4.11: Depth separable convolution

This result is the lightweight deep neural network. The MobileNet model architecture is shown in Fig.4.12.

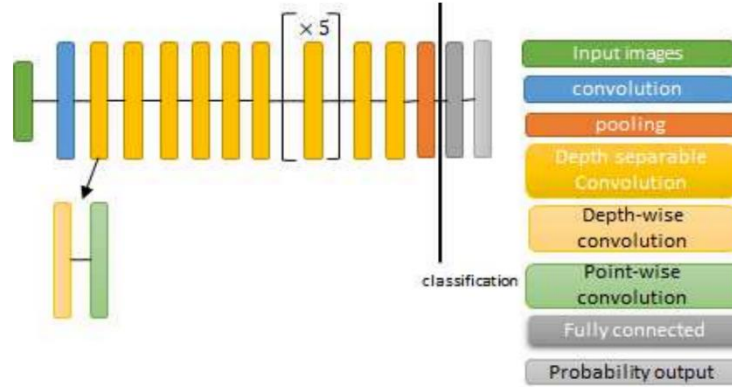


Fig: 4.12: Mobilenet architecture

C. Classification

One of the most common algorithms and methods in the field of classification is the SVM. In the SVM algorithm, each data sample as a point in n-dimensional space on the data scatter plot (n is the number of feature samples). The value of feature is attribute sample, and determines one of the point coordinate components in the diagram. The classification of the different data is accomplished by drawing a straight line. Actually, the SVM algorithm is the boundary that separates the dataset in the best way. The advantage of SVM is its effectiveness in high dimensional space and its ability in solving the over-fitting problem. SVM can separate the input data linearly and nonlinearly. This algorithm uses kernel function for inseparable problems and plot the data in a better viewing space. The most used kernel is described in equation (4.2) - (4.4) Which are linear kernel, Gaussian radial basis function (RBF) kernel and polynomial kernel respectively. The equations of these kernel functions are given as below

$$k(\vec{x}_i, \vec{x}_j) = \vec{x}_i \cdot \vec{x}_j \text{-----equation 4.2}$$

$$k(\vec{x}_i, \vec{x}_j) = \exp(-\gamma (\|\vec{x}_i - \vec{x}_j\|)), \text{ for } \gamma > 0 \text{-----equation 4.3}$$

$$k(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + 1)^d \text{-----equation 4.4}$$

SVM classifier have proved strong performance in flower classification tasks. In this paper after some experiments, they had achieved the almost best accuracy with the LinearSVC library of sklearn.

4.3 DATASET

The Oxford-102 Flower Dataset provided by Nilsback and Zisserman is selected for experimentation. It consists of 102 types of flowers.



Fig. 4.13: Image samples of 25 species from Oxford-102 Flower.

The flower classes present in that dataset are the most common flower species found in the United Kingdom. A total of 8189 images were present. For each category, the images will range from 40 to 258. This paper proposes a CBIR that should be suitable for both small and large datasets and also supports even fewer complex machines. However, using the complete dataset will require a more complex system. It requires a very advanced GPU. Minimizing the dataset helps reduce the system's complexity. Hence, the dataset is reduced to 2040 images through random selection for experimentation. The reduced dataset consists of 20 images for each flower category. 50 percent of the images are used for training purposes, and the remaining 50 percent of images are used for testing.

4.4 EXPERIMENTAL RESULTS OF EXISTING MODELS

Using the MATLAB software tool, we analyzed the performance of pre-trained networks using the reduced dataset collected from the Oxford-102 Flower Dataset. We had set the maximum number of epochs to a stable value of 6 and the mini-batch size to 10. The image-net dataset contains 1000 categories in which pre-trained networks were trained; in order to apply pre-trained networks to flower classification, we fine-tuned the output of the last classification layer to 102. The pre-trained networks used for experimentation are Alexnet, Googlenet, Mobilenetv2, Nasnetmobile, Squeezenet, Resnet18, Resnet50, Shufflenet and Darknet19.

Table 4.1 Accuracy of Networks

Pre-Trained Network	Accuracy
Alexnet	71.764
Darknet19	85.196
Googlenet	74.215
Squeezenet	63.039
Nasnetmobile	70.098
Mobilenetv2	84.313
Resnet18	84.117
Resnet50	88.431
Shufflenet	80.882

Some better performance results were achieved by the respective pre-trained models: Mobilenetv2, Resnet18, Resnet50, Shufflenet, and Darknet19. The highest level of accuracy is achieved in the resnet50. Resnet50 has a relatively large number of layers. So, in order to achieve better results with fewer layers, we decided to modify the resnet18. Figure 4.14 depicts the specificity performance predicted by resnet50 for each category. Figure 4.15 depicts the resnet18 specificity performance of each category, while Figure 4.16 depicts the darknet19 specificity performance of each category. The specificity results show that the resnet50 predicted the Globe Thistle, Giant White Arum Lily, Mexican Aster, and Barberton Daisy perfectly while the resnet18 failed to predict them at 100% accuracy.

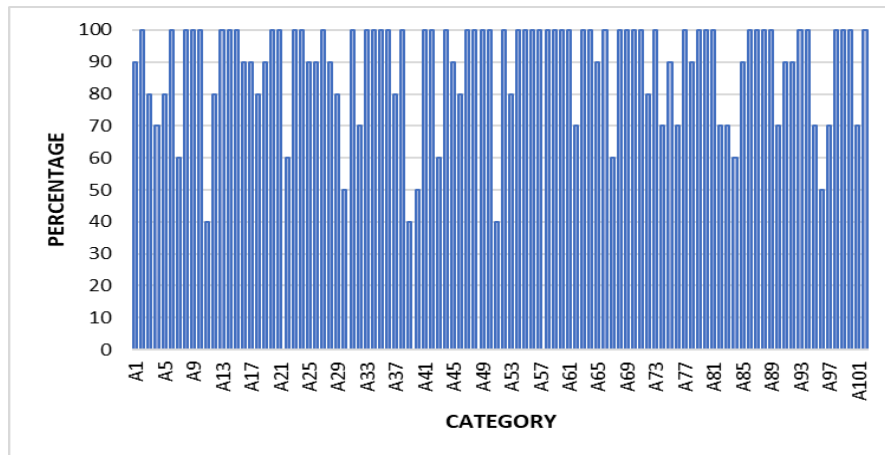


Fig. 4.14: Specificity of individual categories using Resnet50

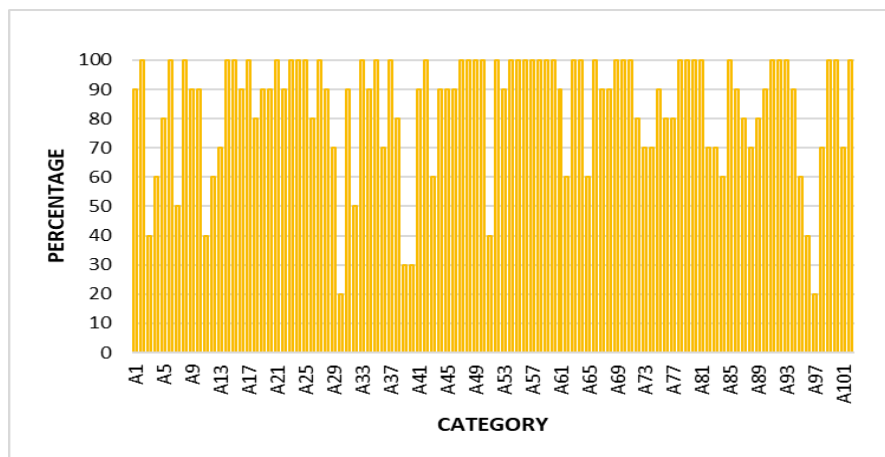


Fig. 4.15: Specificity of individual categories using Resnet18

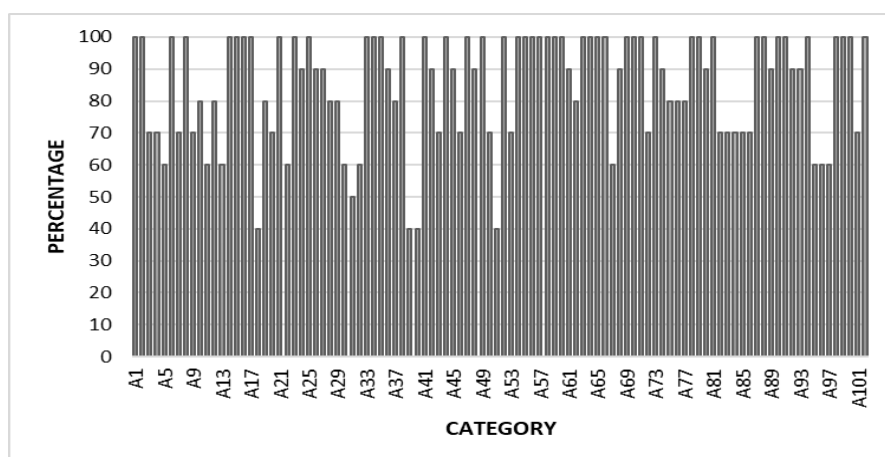


Fig. 4.16: Specificity of individual categories using Darknet19

CHAPTER 5

PROPOSED WORK

For a huge dataset, deeper networks with a large number of layers will give a high-level prediction of images, resulting in better accuracy than the less-layered networks. Deeper networks will extract nearly all of the features from images. In image prediction, networks with fewer layers outperform large layered networks when a small dataset is used for experimentation. Because deeper networks extract more features from the small dataset, overfitting occurs. Despite outperforming the other pre-trained networks in terms of accuracy, resnet50 has a higher layer count than resnet18. We chose to build a CNN with fewer layers, which can outperform large layered networks on both large and small datasets. As a result, the modification work is carried out in resnet18, which is less deep and also employs the feed-forward concept. Chen Jianfei and Zhu Changming developed a resnet model called the improved resnet. The network structure is multi-branch, and its basic blocks are linked via inner connections. Whereas the proposed system represents a network in which the actual resnet18 has some additional layers at the final layers, the additional layers were connected through series connections and no shortcuts were included.

Time consumption as well as difficulties on manual identification of all widely various flower species is cumbersome for even botanical experts. There exist certain challenges in flowers categorization, due to complication of the background of flower images. It is impossible to rely on just a single feature, e.g., color, texture or shape for distinguishing the species of flowers owing to similarity between the different flower categories. In addition, the same species of flowers may be different because of the shape, scale, viewpoint. Convolutional neural networks (CNNs) [1] have been widely utilized in object recognition, including flower image classification. CNNs can well capture significant image features through convolutional layers, without a-priori explicitly defining those features. Former successful CNNs, e.g., AlexNet [2] and VGGNet [3] using naïvely stacking, over-deeply large convolutional layers which consume a vast computation leads to high complexity. Importantly, such over-deep-layer CNNs architecture decrease quality of the model in terms of vanishing gradients as well as model overfitting. ResNet [4] architecture, based on

deep residual-learning convolutional neural network (DRLCNN) has been designed to solve such difficulties.

5.1 RESIDUAL NEURAL NETWORK

The idea of Resnet is to implement residual blocks in the network, which are meant to overcome the problem of vanishing gradient. The vanishing gradient occurs due to the presence of more deep layers, but in resnet the inputs are fed back through a shortcut and get concatenated to the output of respective convolutional blocks. The input size of resnet is $224 \times 224 \times 3$, where 224×224 represents the height and width, and 3 represents the number of color channels (R, G, and B). The dimensions of the input images were 2D. There will be two types of blocks in resnet, namely residual identity blocks and residual convolutional blocks. In an identity block, the inputs will be passed through a set of convolution layers. Each convolution layer will be followed by a batch normalization layer and a relu layer, but for the last convolution layer, the relu layer will be present after the concatenation layer. The input will also get fed directly to the concatenation layer via a shortcut. Figure 5.1 clearly explains the structure of residual identity blocks present in resnets. In a residual convolution block, the input will be passed to a set of convolution layers, which will be followed by a batch normalization layer and a Relu layer. The shortcut will not be directly connected to the concatenation layer. Instead, it will be passed to the convolution layer, followed by a batch normalization layer, and a relu layer will be present after concatenation. Figure 5.2 shows the structure of the residual convolution block.

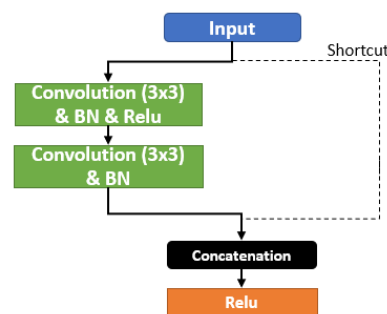


Fig. 5.1: Residual identity block

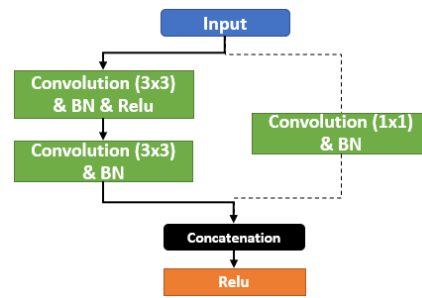


Fig. 5.2: Residual convolutional block

5.2 Modified_Resnet-1

A set of layers termed "modification set-1" is added in between the last residual identity block and the global average pooling layer of resnet18 as shown in Figure 5.3. This network is named modified_resnet-1. The modification set-1 will have two convolution layers, each consisting of 1024 filters. Every convolutional layer is followed by a batch normalization layer and a relu layer. The first convolution layer will have filters of size 7 x 7, and the second layer will have filters of size 5 x 5.

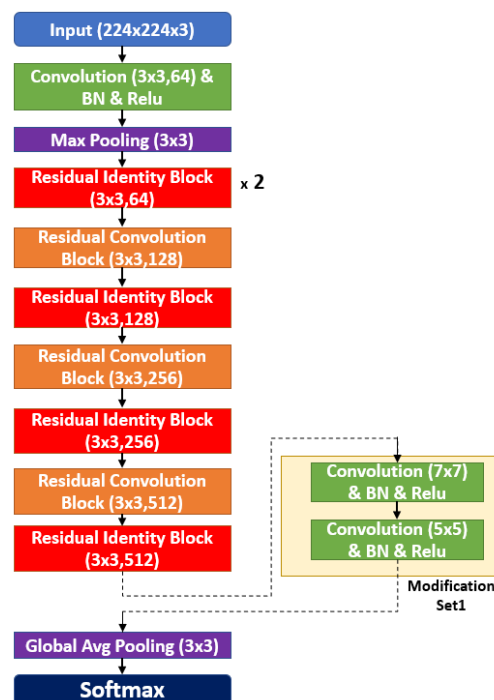


Fig. 5.3: Modified_Resnet-1

Increasing the number of filters by adding some convolution layers at the final layers will help with feature extraction. Accordingly, the addition of modification set-1 improved the performance of the actual resnet18.

5.3 Modified_Resnet-2

The modification set-2 contains a total of 9 layers, including the batch normalization layers and the relu layers. Each convolution layer in modification set-2 consists of 1024 filters. The first convolution layer in modification set-2 has filters of size 7 x 7, the second layer has filters of size 5 x 5, and the third layer will have filters of size 3 x 3. The modification set-2 is added to the resnet18 instead of the modification set-1, as shown in Fig. 5.4. This network is named modified_resnet-2.

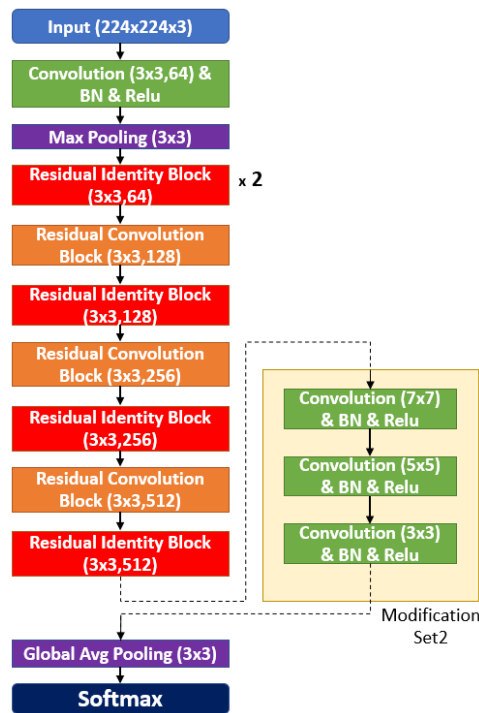


Fig. 5.4: Modified_Resnet-2

5.4 Experimental Results

The addition of modification set-1 achieved an accuracy of 88.82% at the end of epoch 6 for the reduced dataset collected from the respective Oxford-102 flower dataset. The result of modified_resnet-1 is approximately 4.7% greater than the actual pre-trained resnet18 which resulted in an accuracy of 84.11%. The result of modified_resnet-1 is even better than resnet50 by 0.392%, whereas the actual resnet50 resulted in an accuracy of 88.43%. The modified_resnet-1 dominates the

modified_resnet-2 in terms of its performance in the prediction of images. Since modified_resnet-2 has one more convolution layer, it gave rise to overfitting. Anyhow, the modified_resnet-2 achieved an accuracy of 88.62 percent, which is also greater than the resnet18 by 4.4% and better than the resnet50 by 0.196%. The modified_resnet-1 achieved the greatest accuracy when compared to the modified_resnet-2, resnet18, and resnet50. The specificity results show that the modified_resnet-1 and the modified_resnet-2 predicted the images of the Globe flower, Stemless Gentian, and Thorn apple perfectly, whereas the actual resnet18 and resnet50 failed to predict 100% of the images of them. Only modified_resnet-1 successfully predicted all the test images of English Marigold and Primula.

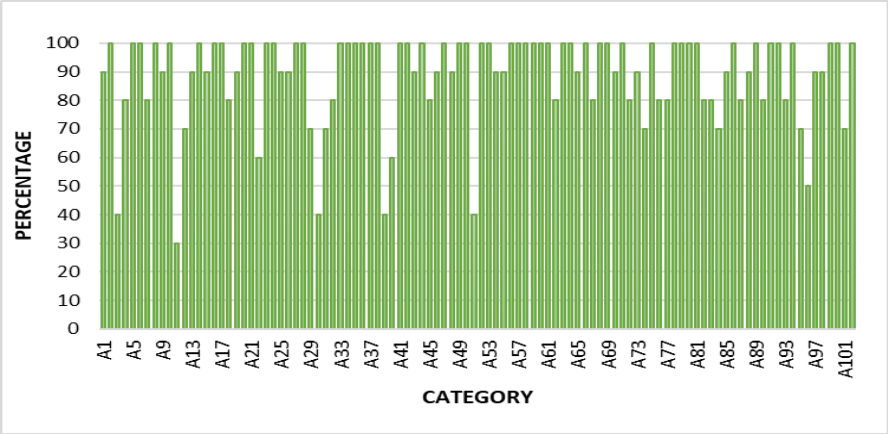


Fig. 5.5: Specificity of individual categories using Modified_resnet-1

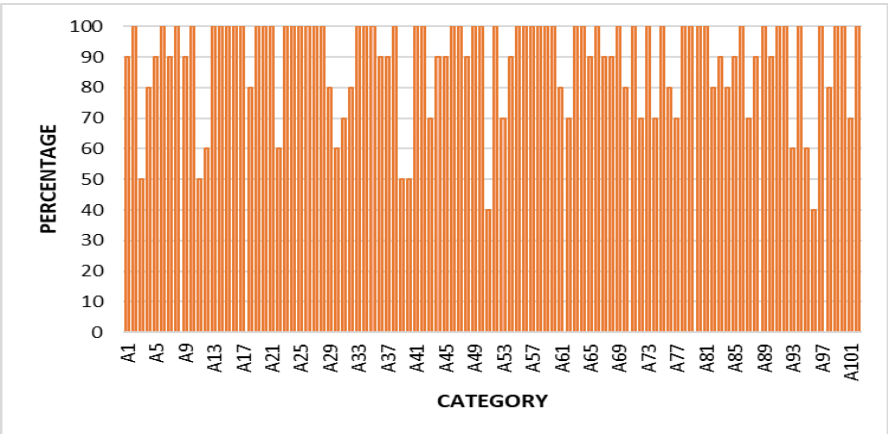


Fig. 5.6: Specificity of individual categories using Modified_resnet-2

For each category, a total of 10 images were tested. Table 2 clearly explains the number of images predicted successfully out of 10 by the existing resnet18 and

resnet50 and by the proposed modified_resnet-1 and modified_resnet-2. As for the 102 flower categories, both the existing and proposed methods correctly detected all the test images for 30 of them, including the Hard-Leaved Pocket Orchid and the Blackberry Lily. Out of 10 images of English Marigolds, the existing models predicted only eight, while all ten were successfully predicted by modified_resnet-1. The modified_resnet-2 found nine images of English Marigold. Both existing models detected six Sword Lily images, whereas modified_resnet-1 correctly predicted nine images. The modified_resnet-2 predicted seven images of Sword Lilies. The images of Stemless Gentian predicted by existing models were nine, whereas the proposed systems predicted 10. From the table, it is seen that the modified_resnet-1 and modified_resnet-2 are more efficient and outperform the existing models.

Table 5.1 Comparision of the Proposed system over the Existing System

Models	Accuracy
Resnet18	84.117
Resnet50	88.431
Modified_Resnet-1	88.823
Modified_Resnet-2	88.627

Table 5.1 shows that Modified_resnet-1 achieves the highest level of accuracy. Comparing the overall accuracy values of existing and proposed systems indicates that modified_resnet-1 and modified_resnet-2 will be more efficient for CBIR applications than the existing CNNs.

With the development of technology, we can shoot the images of flowers more and more conveniently through a variety of electronic equipments, but to distinguish the types of flowers is a difficult thing, even for humans — certainly harder than discriminating a car from a bicycle. At present, most researches of classification are based on the low-level visual aspect (such as color, shape, or texture). They can't be used in flower images classification due to the large intra-class variations, small inter-class differences and the complex background inside natural scenes of flower images. In general, the image requires a preprocessing operation. Three steps are then performed in turn, including feature extraction,

representation of feature vector and design of classifier. Image preprocessing can eliminate unrelated information, filter out interference and improve the detectability of the information, thereby it is very beneficial for image classification. Sum-Product Networks (Softmaxs) have attracted much attention as a special type of deep neural network with clear semantics and a special type of probabilistic graphical model for which inference is always tractable. Compared to general graphical models (such as Bayesian networks, Markov networks), Softmaxs have a higher degree of accuracy and can integrate all the characteristics into a model without requiring approximate reasoning. Thus, it is practicable to use Softmax as a classifier for flower image classification. In this paper, we present a novel approach based on Softmax for flower images classification. The combination of image pre-processing based on aspect ratio preserving and an efficient feature extraction method is created for image classification by using Softmax classifier.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

The convolutional neural network used in this work is more suitable and efficient for feature extraction than the conventional method of feature extraction. This work uses the Oxford-102 Flower dataset. To reduce the system complexity, several randomly selected 2040 images have been used instead of the complete 8189 images. For experimentation, the maximum number of epochs is assigned to 6. Among the pre-trained existing models, the ResNet50 resulted in the highest accuracy of 88.43%. The ResNet50 is too large because it has more layers. Thus, this paper focuses on developing a CNN that should be suitable for both large and small datasets. Because of the limited number of layers in ResNet18, the modification is done in ResNet18 to outperform large layered networks. By modifying ResNet18 twice, two CNNs were created. Both modified_resnet-1 and modified_resnet-2 performed better than the existing resnet models 18 and 50. The modified_resnet-1 in this paper achieved the best result of 88.82% accuracy when compared to the existing resnet18 and the resnet50. This proposed CNN will be more suitable for CBIR applications than conventional CNNs.

REFERENCES

- [1] Faeze Sadati, Behrooz Rezaie, “An Improved Image Classification Based In Feature Extraction From Convolutional Neural Network: Application To Flower Classification”, 12th International Conference on Information and Knowledge Technology (IKT), 2021.
- [2] Xudong Wei, Hongyu Zhang, Changkai Shi, Xinyue Yang, Huan Han, Bo Li, “A Lightweight Flower Classification Model Based on Improved Knowledge Distillation”, IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), 2022.
- [3] Muhammed YILDIRIM, Ahmet Çınar, Emine CENGİL, “Classification of flower species using CNN models, Subspace Discriminant, and NCA”, International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), 2021.
- [4] Dr. Shantala Giraddi, Dr. Shivanand Seeri, Dr. P S Hiremath, Dr. Jayalaxmi G.N, “Flower Classification using Deep Learning models”, International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE 2020).
- [5] Yong Wu, Xiao Qin, Yonghua Pan, Changan Yuan, “Convolution Neural Network based Transfer Learning for Classification of Flowers”, IEEE 3rd International Conference on Signal and Image Processing, 2018.
- [6] Sandip Desai, Chetan Gode, Punit Fulzele, “Flower Image Classification Using Convolutional Neural Network”, 2022 First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT).
- [7] Faisal Ridwan Siregar, Wikky Fawwaz Al Maki, “Hybrid Method for Flower Classification in High Intra-class Variation”, 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 2020.
- [8] Pheeraphat Sarikabuta, Siriporn Supratid, “Impacts of Layer Sizes in Deep Residual-Learning Convolutional Neural Network on Flower Image Classification with Different class sizes”, 2022 International Electrical Engineering Congress (IEECON).

- [9] Xiaoling Xia, Cui Xu, "Inception-v3 for Flower Classification", 2nd International Conference on Image, Vision and Computing, 2017
- [10] Fei Hu, Fuguang Yao, Changjiu Pu, "Learning Salient Features for Flower Classification Using Convolutional Neural Network", 2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIS).
- [11] Emine CENGİL, Ahmet ÇINAR, "Multiple Classification of Flower Images Using Transfer Learning", 2019 International Artificial Intelligence and Data Processing Symposium (IDAP).
- [12] Visual Geometry Group, Flower Datasets Home Page, (2008) <https://www.robots.ox.ac.uk/~vgg/data/flowers/102/> (accessed December 12, 2022).
- [13] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.C. "MobileNetV2: Inverted Residuals and Linear Bottlenecks." In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4510-4520). IEEE.
- [14] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition", 2016 IEEE conference on computer vision and pattern recognition, pp. 770-778.
- [15] Redmon, Joseph. "Darknet: Open Source Neural Networks in C", <https://pjreddie.com/darknet>.
- [16] Zhang, Xiangyu, Xinyu Zhou, Mengxiao Lin, and Jian Sun. "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices", 2017 arXiv preprint arXiv:1707.01083v2.
- [17] Iandola, Forrest N., Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size", Submitted the Preprint on November 4, 2016. <https://arxiv.org/abs/1602.07360>
- [18] Zoph, Barret, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. "Learning Transferable Architectures for Scalable Image Recognition", 2017 arXiv preprint arXiv:1707.07012 2, no. 6.

- [19] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks", 2012 Advances in neural information processing systems.
- [20] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions", 2015 IEEE conference on computer vision and pattern recognition, pp. 1-9.
- [21] Chen Jianfei, Chen Jianfei, "Research on Image Recognition Based on Improved ResNet", 2020 IEEE 6th International Conference on Computer and Communications.
- [22] M.A. Muthiah, E. Logashamugam, N.M. Nandhitha, "Performance evaluation of Googlenet, Squeezenet, and Resnet50 in the classification of Herbal images", International Journal of Engineering Trends and Technology, Volume 69 Issue 3, 229-232, March 2021.
- [23] M.A. Muthiah, E. Logashanmugam, "Feasibility of conventional neural networks for content-based image retrieval in big data", 2021 International Journal of Public Sector Performance Management.
- [24] Muthiah.M.A, Logashanmugam.E, Nandhitha.N.M, "Feasibility of Convolutional Neural Networks (CNN) for Content Based Image Retrieval of Images from Hadoop" 2021 In book: Smart Intelligent Computing and Communication Technology.

APPENDIX

A. RESEARCH PAPER

A Novel and Efficient CBIR using CNN for Flowers

Subash. S. I

Department of Electronics and
Communication Engineering
Sathyabama University
Chennai, India
subash230102@gmail.com

Muthiah. M. A. M.Tech
Assistant Professor, Department of
Electronics and
Communication Engineering
Sathyabama University
Chennai, India
muthiah.m.a@outlook.com

Dr. N. Mathan. M. Tech., Ph.D
Assistant Professor, Department of
Electronics and
Communication Engineering
Sathyabama University
Chennai, India
mathan.ntml@gmail.com

Abstract— Image processing is vital to extract the required data from images. Machine learning is an efficient tool used for penetration in most of the classification and identification tasks performed by a computer. This project proposes the identification of a flower after the classification of flower images using a successful artificial intelligence tool named the Convolutional Neural Network (CNN). Models similar to this project have been used in most search engines for a long time, but CBIR (content-based image retrieval) still runs with less accuracy and produces outputs with fewer specifications due to the use of convolutional feed-forward networks for image retrieval. System performance depends a lot on the drawn-out features extracted from images. So, it is required to develop a CBIR system that retrieves similar images without explicit feature extraction and classification by using CNN, which accepts images as input. For experimentation, images from the Oxford-102 flower dataset are used.

Keywords— convolutional neural network; content-based image retrieval; image processing; machine learning

I. INTRODUCTION

The number of flower plant species identified up to date is nearly 3,69,000. Therefore, it is necessary to identify and classify them. If the work is to be done manually, it will be a tougher task, but if we go up with the help of machine learning, we can achieve better results more easily. Although similar work has been done regarding flower classification using CNNs, the performance of pre-trained networks is not satisfactory. The flower classification task makes it much tougher because the shape, color, size, and petal position of each flower are unique, and for some, they might be similar. In some species, the petal position changes depending on the conditions; for example, when it is a bud, the petals are closed, and when it blooms, the petals are open. It also depends on the surrounding temperature and climate. Every pre-trained convolutional neural network had a different structure because it was developed for several different applications and to overcome the drawbacks of previous networks. So, the work is to modify the pre-trained networks, which would give better results than the actual ones.

II. NEURAL NETWORK STRUCTURE

In general, a deep neural network will have layers like a convolution layer, a normalization layer, an activation function, a pooling layer, a fully connected layer, and a classifier. The layers are responsible for the functioning of a network. The convolution layers will have a number of filters that will extract high-level features such as edges and curvatures from the input images. The size of the filters will be $n \times n$. It will have padding options that we must assign, as well as the stride size. In a network, the first set of convolutional layers is used to extract

low-level features, and the subsequent convolutional layers are for extracting high-level features. The normalization layer will normalize the range of input values because if two or more inputs in a network have different ranges of values, the weights will become different, resulting in a vanishing gradient. There are some popular normalization layers, like Cross-Channel Normalization and Batch Normalization. The batch normalization will normalize the range of values of the outputs of each layer. The activation function is to break the linearity in networks and introduce non-linearity. The pooling layer is used to downsample the convolved features after convolution; this will reduce the dimension of images and extract dominant features. The Fully Connected layer will have the weights; it is a combination of high-level features in a non-linear way. Classifier, the last layer in deep neural networks, will classify the images into the mentioned categories.

III. RELATED WORK

Several works on flower classification have been published. Faeze Sadati and Behrooz Rezaie [1] compared the pre-trained networks Resnet101, Inceptionv3, Densenet201, and Mobilenetv3 and combined the networks to create a new network that outperforms those pre-defined networks. Xudong Wei et al. [2] presented a lightweight flower classification system based on a CNN created by modifying a pre-trained Mobilenetv3 model. They had undergone several modifications to overcome the actual Mobilenetv3. Muhammed YILDIRM et al. [3] presented a flower classification system in which they used a dataset of 10 flower categories, compared pre-trained networks like Alexnet, Mobilenetv2, and Efficientnetb0, and modified them as well. Dr. Shantala Giraddi et al. [4] presented a CNN that consists of four convolutional layers and a fine-tuned VGG-16. They compared it to an actual, pre-trained Vgg16. Yong Wu et al. [5] confirmed that the transfer learning approach over pre-trained networks is giving better results than random initialization models. Sandip Desai et al. [6] used several machine-learning techniques and achieved higher accuracy in pre-trained Vgg19. They used the Oxford-17 Flower dataset for experimentation. Faisal Ridwan Siregar and Wiky Fawwaz Al Maki [7] compared the performance of three popular classifiers: Random Forest, SVM Polynomial Kernel, and SVM Linear Kernel. Random Forest has the best performance. They achieved 88.74% accuracy in their presented system. Pheeraphat Sarikabuta and Siriporn Supratid [8] compared the pre-trained Resnet17, Resnet35, and Resnet50. They achieved the highest result in Resnet 35. Xiaoling Xia et al. [9] experimented with the pre-trained Inception v3 model over Tensorflow. They used both the Oxford-17 flower dataset and the Oxford-102 flower dataset. Fei Hu et al. [10] tested the performance of CNN in the presence and absence of a dropout layer. They also used flower classification to compare the activation functions: sigmoid and

relu. Emine CENGIL and Ahmet CINAR [11] compared the performances of Alexnet, Resnet18, Googlenet, Densenet201, and Vgg16. When comparing the viability of pre-trained networks for classifying herbs, A. Muthiah et al. [22] found that Resnet50 is extremely sensitive to herbal pictures. PNN is effective for DWT characteristics, according to research on CBIR applications with conventional neural networks [23]. In CBIR for flower categorization, Densenet201 outperforms conventional networks with the maximum level of accuracy [24]. The experimentation process was undertaken in Matlab software.

IV. DATASET

The Oxford-102 Flower Dataset [12] provided by Nilsback and Zisserman is selected for experimentation. It consists of 102 types of flowers. The flower classes present in that dataset are the most common flower species found in the United Kingdom. A total of 8189 images were present. For each category, the images will range from 40 to 258. This paper proposes a CBIR that should be suitable for both small and large datasets and also supports even fewer complex machines. However, using the complete dataset will require a more complex system. It requires a very advanced GPU. Minimizing the dataset helps reduce the system's complexity. Hence, the dataset is reduced to 2040 images through random selection for experimentation. The reduced dataset consists of 20 images for each flower category. 50 percent of the images are used for training purposes, and the remaining 50 percent of images are used for testing.



Fig. 1. Image samples of 25 species from Oxford-102 Flower.

V. EXISTING WORK ANALYSIS

Using the MATLAB software tool, we analyzed the performance of pre-trained networks using the reduced dataset collected from the Oxford-102 Flower Dataset [12]. We had set the maximum number of epochs to a stable value of 6 and the mini-batch size to 10. The image-net dataset contains 1000 categories in which pre-trained networks were trained; in order to apply pre-trained networks to flower classification, we finetuned the output of the last classification layer to 102. The pre-trained networks used for experimentation are Alexnet [19], Googlenet [20], Mobilenetv2 [13], Nasnetmobile [18],

Squeezenet [17], Resnet18 [14], Resnet50 [14], Shufflenet [16] and Darknet19 [15]. Some better performance results were achieved by the respective pre-trained models: Mobilenetv2 [13], Resnet18 [14], Resnet50 [14], Shufflenet [16], and Darknet19 [15].

TABLE 1. ACCURACY OF NETWORKS

Pre-Trained Network	Accuracy
Alexnet	71.764
Darknet19	85.196
Googlenet	74.215
Squeezenet	63.039
Nasnetmobile	70.098
Mobilenetv2	84.313
Resnet18	84.117
Resnet50	88.431
Shufflenet	80.882

The highest level of accuracy is achieved in the resnet50 [14]. Resnet50 [14] has a relatively large number of layers. So, in order to achieve better results with fewer layers, we decided to modify the resnet18 [14]. Figure 2 depicts the specificity performance predicted by resnet50 for each category. Figure 3 depicts the resnet18 specificity performance of each category, while Figure 4 depicts the darknet19 specificity performance of each category. The specificity results show that the resnet50 [14] predicted the Globe Thistle, Giant White Arum Lily, Mexican Aster, and Barberton Daisy perfectly while the resnet18 [14] failed to predict them at 100% accuracy.

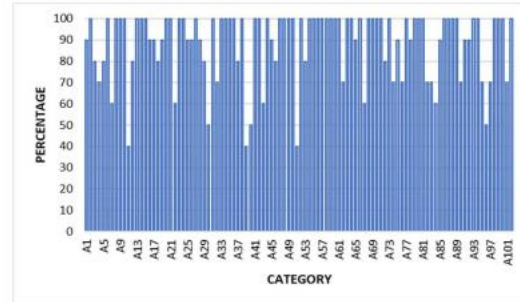


Fig. 2. Specificity of individual categories using Resnet50

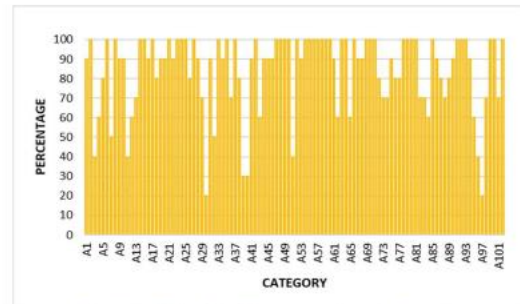


Fig. 3. Specificity of individual categories using Resnet18

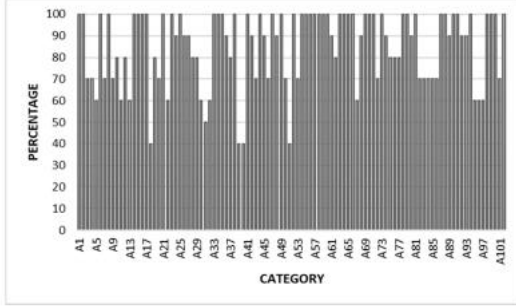


Fig. 4. Specificity of individual categories using Darknet19

VI. PROPOSED METHOD

For a huge dataset, deeper networks with a large number of layers will give a high-level prediction of images, resulting in better accuracy than the less-layered networks. Deeper networks will extract nearly all of the features from images. In image prediction, networks with fewer layers outperform large layered networks when a small dataset is used for experimentation. Because deeper networks extract more features from the small dataset, overfitting occurs. Despite outperforming the other pre-trained networks in terms of accuracy, resnet50 [14] has a higher layer count than resnet18 [14]. We chose to build a CNN with fewer layers, which can outperform large layered networks on both large and small datasets. As a result, the modification work is carried out in resnet18 [14], which is less deep and also employs the feed-forward concept. Chen Jianfei and Zhu Changming [21] developed a resnet model called the improved resnet. The network structure is multi-branch, and its basic blocks are linked via inner connections. Whereas the proposed system represents a network in which the actual resnet18 [14] has some additional layers at the final layers, the additional layers were connected through series connections and no shortcuts were included.

A. Residual Neural Network

The idea of Resnet [14] is to implement residual blocks in the network, which are meant to overcome the problem of vanishing gradient. The vanishing gradient occurs due to the presence of more deep layers, but in resnet [14] the inputs are fed back through a shortcut and get concatenated to the output of respective convolutional blocks. The input size of resnet [14] is $224 \times 224 \times 3$, where 224×224 represents the height and width, and 3 represents the number of color channels (R, G, and B). The dimensions of the input images were 2D. There will be two types of blocks in resnet [14], namely residual identity blocks and residual convolutional blocks. In an identity block, the inputs will be passed through a set of convolution layers. Each convolution layer will be followed by a batch normalization layer and a relu layer, but for the last convolution layer, the relu layer will be present after the concatenation layer. The input will also get fed directly to the concatenation layer via a shortcut. Figure 5 clearly explains the structure of residual identity blocks present in resnets [14]. In a residual convolutional block, the input will be passed to a set of convolution layers, which will be followed by a batch normalization layer and a Relu layer. The shortcut will not be directly connected to the concatenation layer. Instead, it will be passed to the convolution layer, followed by a batch normalization layer, and a relu layer will be present after concatenation. Figure 6 shows the structure of the residual convolution block.

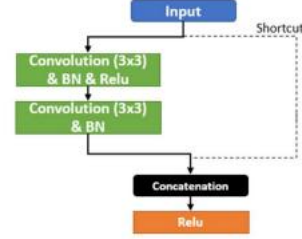


Fig. 5. Residual identity block

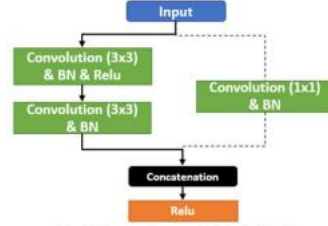


Fig. 6. Residual convolutional block

B. Modified_Resnet-1

A set of layers termed "modification set-1" is added in between the last residual identity block and the global average pooling layer of resnet18 [14] as shown in Figure 7. This network is named modified_resnet-1. The modification set-1 will have two convolution layers, each consisting of 1024 filters. Every convolutional layer is followed by a batch normalization layer and a relu layer. The first convolution layer will have filters of size 7×7 , and the second layer will have filters of size 5×5 .

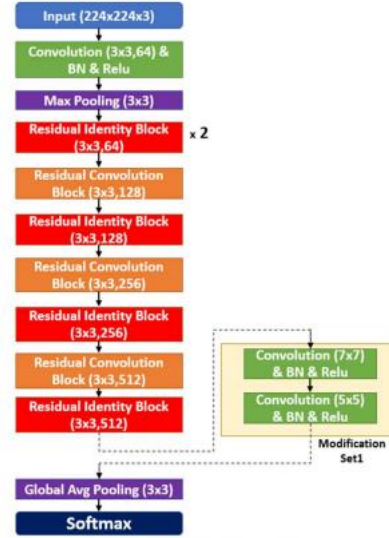


Fig. 7. Modified_Resnet-1

Increasing the number of filters by adding some convolution layers at the final layers will help with feature

extraction. Accordingly, the addition of modification set-1 improved the performance of the actual resnet18 [14].

C. Modified_Resnet-2

The modification set-2 contains a total of 9 layers, including the batch normalization layers and the relu layers. Each convolution layer in modification set-2 consists of 1024 filters. The first convolution layer in modification set-2 has filters of size 7 x 7, the second layer has filters of size 5 x 5, and the third layer will have filters of size 3 x 3. The modification set-2 is added to the resnet18 [14] instead of the modification set-1, as shown in Fig. 8. This network is named modified_resnet-2.

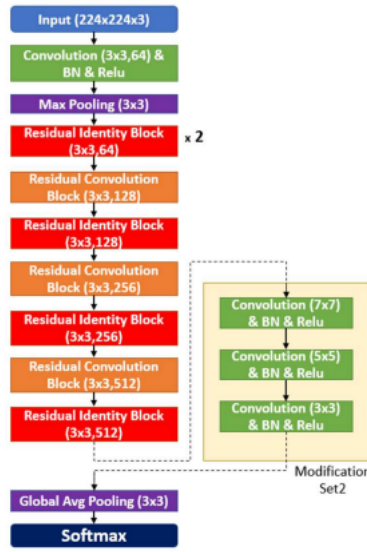


Fig. 8. Modified_Resnet-2

VII. EXPERIMENTAL RESULTS

The addition of modification set-1 achieved an accuracy of 88.82% at the end of epoch 6 for the reduced dataset collected from the respective Oxford-102 flower dataset [12]. The result of modified_resnet-1 is approximately 4.7% greater than the actual pre-trained resnet18 [14] which resulted in an accuracy of 84.11%. The result of modified_resnet-1 is even better than resnet50 [14] by 0.392%, whereas the actual resnet50 [14] resulted in an accuracy of 88.43%. The modified_resnet-1 dominates the modified_resnet-2 in terms of its performance in the prediction of images. Since modified_resnet-2 has one more convolution layer, it gave rise to overfitting. Anyhow, the modified_resnet-2 achieved an accuracy of 88.62 percent, which is also greater than the resnet18 [14] by 4.4% and better than the resnet50 [14] by 0.196%. The modified_resnet-1 achieved the greatest accuracy when compared to the modified_resnet-2, resnet18 [14], and resnet50 [14]. The specificity results show that the modified_resnet-1 and the modified_resnet-2 predicted the images of the Globe flower, Stemless Gentian, and Thorn apple perfectly, whereas the actual resnet18 [14] and resnet50 [14] failed to predict 100% of

the images of them. Only modified_resnet-1 successfully predicted all the test images of English Marigold and Primula.

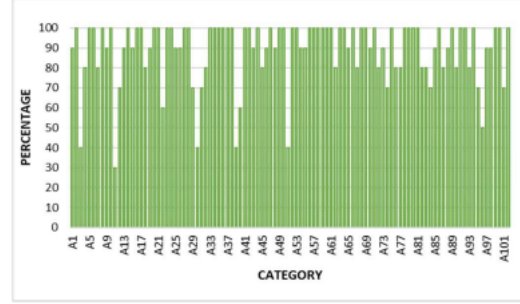


Fig. 9. Specificity of individual categories using Modified_resnet-1

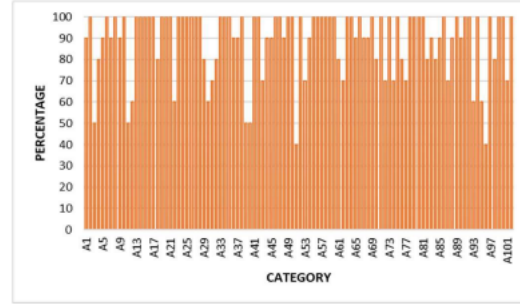


Fig. 10. Specificity of individual categories using Modified_resnet-2

TABLE 2. A FEW SAMPLES OF PREDICTED RESULTS BY THE EXISTING AND PROPOSED MODELS

SN	Category	Resnet 18	Resnet 50	Modified-Resnet-1	Modified-Resnet-2
A1	Pink Primrose	9	9	9	9
A4	Sweet Pea	6	7	8	8
A5	English Marigold	8	8	10	9
A7	Moon Orchid	5	6	8	9
A10	Globe Thistle	9	10	10	10
A16	Globe-Flower	9	9	10	10
A19	Balloon Flower	9	9	9	10
A24	Red Ginger	10	10	10	10
A28	Stemless Gentian	9	9	10	10
A34	Mexican Aster	9	10	10	10
A38	Great Masterwort	8	10	10	10
A43	Sword Lily	6	6	9	7
A46	Wallflower	9	8	9	10
A49	Oxeye Daisy	10	10	10	10
A56	Bishop of Llandaff	10	10	10	10
A62	Japanese Anemone	6	7	8	7

A71	Gazania	10	10	10	10
A75	Thorn Apple	9	9	10	10
A78	Lotus	10	9	10	10
A82	Clematis	7	7	8	8
A84	Columbine	6	6	7	8
A91	Hippeastrum	10	9	10	10
A97	Mallow	2	7	9	10
A10	Blackberry	10	10	10	10
2	Lily	10	10	10	10

For each category, a total of 10 images were tested. Table 2 clearly explains the number of images predicted successfully out of 10 by the existing resnet18 [14] and resnet50 [14] and by the proposed modified_resnet-1 and modified_resnet-2. As for the 102 flower categories, both the existing and proposed methods correctly detected all the test images for 30 of them, including the Hard-Leaved Pocket Orchid and the Blackberry Lily. Out of 10 images of English Marigolds, the existing models predicted only eight, while all ten were successfully predicted by modified_resnet-1. The modified_resnet-2 found nine images of English Marigold. Both existing models detected six Sword Lily images, whereas modified_resnet-1 correctly predicted nine images. The modified_resnet-2 predicted seven images of Sword Lilies. The images of Stemless Gentian predicted by existing models were nine, whereas the proposed systems predicted 10. From the table, it is seen that the modified_resnet-1 and modified_resnet-2 are more efficient and outperform the existing models.

TABLE 3. COMPARISON OF THE PROPOSED SYSTEM OVER THE EXISTING SYSTEM

Models	Accuracy
Resnet18	84.117
Resnet50	88.431
Modified_Resnet-1	88.823
Modified_Resnet-2	88.627

Table 3 shows that Modified_resnet-1 achieves the highest level of accuracy. Comparing the overall accuracy values of existing and proposed systems indicates that modified_resnet-1 and modified_resnet-2 will be more efficient for CBIR applications than the existing CNNs.

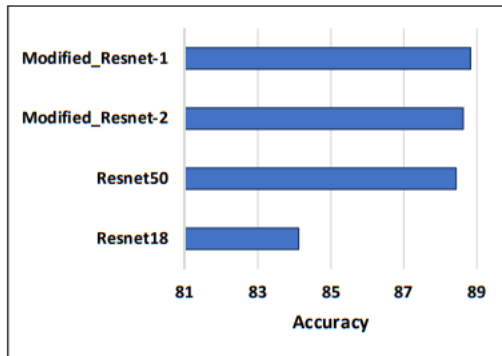


Fig. 11. Accuracy Comparison by Bar Graph

VIII. CONCLUSION

The convolutional neural network used in this work is more suitable and efficient for feature extraction than the conventional method of feature extraction. This work uses the Oxford-102 Flower dataset [12]. To reduce the system complexity, several randomly selected 2040 images have been used instead of the complete 8189 images. For experimentation, the maximum number of epochs is assigned to 6. Among the pre-trained existing models, the ResNet50 [14] resulted in the highest accuracy of 88.43%. The ResNet50 [14] is too large because it has more layers. Thus, this paper focuses on developing a CNN that should be suitable for both large and small datasets. Because of the limited number of layers in ResNet18 [14], the modification is done in ResNet18 [14] to outperform large layered networks. By modifying ResNet18 [14] twice, two CNNs were created. Both modified_resnet-1 and modified_resnet-2 performed better than the existing resnet models 18 and 50. The modified_resnet-1 in this paper achieved the best result of 88.82% accuracy when compared to the existing resnet18 [14] and the resnet50 [14]. This proposed CNN will be more suitable for CBIR applications than conventional CNNs.

REFERENCES

- [1] Faeze Sadati, Behrooz Rezaei, "An Improved Image Classification Based In Feature Extraction From Convolutional Neural Network: Application To Flower Classification", 12th International Conference on Information and Knowledge Technology (IKT), 2021.
- [2] Xudong Wei, Hongyu Zhang, Changkai Shi, Xinyue Yang, Huan Han, Bo Li, "A Lightweight Flower Classification Model Based on Improved Knowledge Distillation", IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), 2022.
- [3] Muhammed YILDIRIM, Ahmet Çınar, Emine CENGİL, "Classification of flower species using CNN models, Subspace Discriminant, and NCA", International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), 2021.
- [4] Dr. Shantala Giraddi, Dr. Shivanand Seeri, Dr. P. S. Hiremath, Dr. Jayalaxmi G.N., "Flower Classification using Deep Learning models", International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE 2020).
- [5] Yong Wu, Xiao Qin, Yonghua Pan, Changan Yuan, "Convolution Neural Network based Transfer Learning for Classification of Flowers", IEEE 3rd International Conference on Signal and Image Processing, 2018.
- [6] Sandip Desai, Chetan Gode, Punit Fulzele, "Flower Image Classification Using Convolutional Neural Network", 2022 First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT).
- [7] Faisal Ridwan Siregar, Wikky Fawwaz Al Maki, "Hybrid Method for Flower Classification in High Intra-class Variation", 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 2020.
- [8] Pheeraphat Sarikabuta, Siriporn Supratid, "Impacts of Layer Sizes in Deep Residual-Learning Convolutional Neural Network on Flower Image Classification with Different class sizes", 2022 International Electrical Engineering Congress (IEECON).
- [9] Xiaoling Xia, Cui Xu, "Inception-v3 for Flower Classification", 2nd International Conference on Image, Vision and Computing, 2017.
- [10] Fei Hu, Fuguang Yao, Changjiu Pu, "Learning Salient Features for Flower Classification Using Convolutional Neural Network", 2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIS).
- [11] Emine CENGİL, Ahmet ÇINAR, "Multiple Classification of Flower Images Using Transfer Learning", 2019 International Artificial Intelligence and Data Processing Symposium (IDAP).
- [12] Visual Geometry Group, Flower Datasets Home Page, (2008) <https://www.robots.ox.ac.uk/~vgg/data/flowers/102/> (accessed December 12, 2022).
- [13] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.C. "MobileNetV2: Inverted Residuals and Linear Bottlenecks." In 2018

- IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4510-4520). IEEE.
- [14] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition", 2016 IEEE conference on computer vision and pattern recognition, pp. 770-778.
 - [15] Redmon, Joseph. "Darknet: Open Source Neural Networks in C", <https://pjreddie.com/darknet>.
 - [16] Zhang, Xiangyu, Xinyu Zhou, Mengxiao Lin, and Jian Sun. "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices", 2017 arXiv preprint arXiv:1707.01083v2.
 - [17] Iandola, Forrest N., Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size", Submitted the Preprint on November 4, 2016. <https://arxiv.org/abs/1602.07360>
 - [18] Zoph, Barret, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. "Learning Transferable Architectures for Scalable Image Recognition", 2017 arXiv preprint arXiv:1707.07012 2, no. 6.
 - [19] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks", 2012 Advances in neural information processing systems.
 - [20] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions", 2015 IEEE conference on computer vision and pattern recognition, pp. 1-9.
 - [21] Chen Jianfei, Chen Jianfei, "Research on Image Recognition Based on Improved ResNet", 2020 IEEE 6th International Conference on Computer and Communications.
 - [22] M.A. Muthiah, E. Logashanmugam, N.M. Nandhitha, "Performance evaluation of Googlenet, Squeezenet, and Resnet50 in the classification of Herbal images", International Journal of Engineering Trends and Technology, Volume 69 Issue 3, 229-232, March 2021.
 - [23] M.A. Muthiah, E. Logashanmugam, "Feasibility of conventional neural networks for content-based image retrieval in big data", 2021 International Journal of Public Sector Performance Management.
 - [24] Muthiah.M.A, Logashanmugam.E , Nandhitha.N.M, "Feasibility of Convolutional Neural Networks (CNN) for Content Based Image Retrieval of Images from Hadoop" 2021 In book: Smart Intelligent Computing and Communication Technology.