AN TRANSFER LEARNING APPROACH FOR IMAGE CLASSIFICATION USING BINARY IMAGE SEGMENTATION ON LIMTED DATASET

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY

IN

INFORMATION SYSTEMS

Submitted By:

B N DEEPANKAN

(2K17/ISY/04)

Under the supervision of
Ms. Ritu Agarwal

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

JUNE, 2019

CANDIDATE'S DECLARATION

I, B N Deepankan, Roll No. 2K17/ISY/04 student of M.Tech Information Systems, hereby declare that the project Dissertation titled "An Transfer Learning Approach For Image Classification Using Binary Image Segmentation On Limited Dataset" which is submitted by me to the Department of Information Technology, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi B N Deepankan

Date:

CERTIFICATE

I hereby certify that the Project Dissertation titled "An Transfer Learning Approach For Image Classification Using Binary Image Segmentation On Limited Dataset" which is submitted by B N Deepankan, Roll No 2K17/ISY/04 Information Technology, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi **Ms Ritu Agarwal**

Date: **SUPERVISOR**

ACKNOWLEDGEMENT

I express my gratitude to my major project guide Ms. Ritu Agarwal, Assistant Professor, lT Dept., Delhi Technological University, for the valuable support and guidance she provided in making this major project. It is my pleasure to record my sincere thanks to my respected guide for his constructive criticism and insight without which the project would not have shaped as it has.

I humbly extend my words of gratitude to other faculty members of this department for providing their valuable help and time whenever it was required.

B N Deepankan

Roll No. 2K17/ISY/04

M.Tech (Information Systems)

E-mail: bn.deepankan@gmail.com

ABSTRACT

Image classification has become a part of our daily routine whether it is classifying between traffic signals or different types of species. However, to differentiate between similar texture and shapes is a difficult task with a naked eye. Latest advancements in the field of computer vision can make this task of image classification easier with deep learning techniques, especially neural networks.

However, training neural networks require large datasets, otherwise, it cannot give accurate classification. Inspite all the data availability, there are some subjects which lack enough data. Medical images, rare animals species to name a few examples with relatively less number of information. In our experiment, we have taken those animal species datasets with a minimum number of data and achieved higher classification accuracy.

We have examined the various state-of-the-art neural networks like DenseNet and Convolutional Neural Networks that could classify between various animal breeds, and flower species. Furthermore, we compared their results based on accuracy achieved on the test set to determine the most efficient approach. Thus, we could assess which network is most suited for image classification.

Moreover, we proposed a two-phase algorithm which differentiates between multiple image dataset through transfer learning via pre-trained Convolutional Neural Network. Initially, images are automatically segmented with the Fully connected network to allow localization of the subject through minimum bounding box around it. Second, we built a robust convolution neural network fine-tuned with a dense network according to our image datasets. We also proposed novel steps during the training stage to ensure a robust, accurate and real-time classification. Finally, we have evaluated our method on the well-known dog breed dataset, and bird species dataset. The experimental results outclass the earlier methods and achieve an accuracy of 95% to 97% for classifying these datasets.

CONTENTS

List of Figures

List of Tables

List of Symbols, Abbreviations and Nomenclature

VGG – Visual Geometry Group

ResNet – Residual Network

DenseNet – Dense Neural Network

FCN – Fully Connected Network

CNN – Convolutional Neural Network

ReLu - Rectified Linear Unit

RCNN – Region Convolutional Neural Network

YOLO – You Only Look Once

2D – Two Dimensional

3D – Three Dimensional

SVM – Support Vector Machine

DCNN – Deep Convolutional Neural Network

RGB – Red Green Blue

PCA – Principal Component Analysis

ICA – Independent Component Analysis

μ – Mean

σ – Variance

Σ – Summation

Ω - Subset

CHAPTER 1 INTRODUCTION

1.1 IMAGE CLASSIFICATION

Image classification means classifying between subjects using images, the matter of interest. It is performed on a daily basis whether differentiating between a traffic signal or choosing between apple from oranges. Although the process seems to be an easy task for an individual, the machine cannot classify on the go. However, with the latest advancements in deep learning techniques, training the machine for accurate classification is possible.

Besides from minor classification problems such as distinguishing between dogs from cats, analyzing an image for predicting the future outcome became a major field of interest. Computer classification of remotely sensed images, medical images, and acoustic images involves the process of computer program learning the relationship between the data and the information class. As per the medical imaging is concerned most of the images may be used in the detection of tumors or for screening the patients.

In addition to the above-mentioned applications, image classification approach was a great help in solving a wide variety of problems. However, in order to get the correct classification, these problems usually need techniques which are capable of enhancing information for accurate analysis. Some of the few techniques in image processing procedures involve removal of noise or restoration of blurred images. The current area of application of image classification techniques is in solving the problem of machine vision so as to attain higher differentiation results.

Though learning a machine for efficient classification can be conducted it will require a huge dataset to do so. However, for some problems, sufficient data are not available to train the model precisely. In those situations, data augmentation methods such as rotation, scaling and inverting the available images on the dataset to increase classification accuracy can be implemented. Although, it must be taken into account that excessive training can lead to overfitting of data which can cause inaccurate classification to test images.

Apart from simple object classification such as differentiating between black and white, distinguishing between similar objects is a challenging task due to a wide range of matching shapes, color, texture and appearance. Furthermore, images of different subjects in the wild such as subjects in the park usually contain similar surrounding objects like leaves, grass, etc. We have considered a dataset containing 20,835 images of 120 dog breeds [1], another dataset with 6,033 images of 200 bird species [2]. These datasets contain images from similar background moreover, each type is comparable in shape, size, and light. However, manual classification is possible by a trained professional but it can be a tedious task for a pedestrian. Hence, learning a machine for performing such classification can reduce the hassle involved.

In general, there are two ways through which machines can learn image classification:

1.1.1 Unsupervised Learning

In Unsupervised learning, learning happens without a teacher. The training data consisting of {x}, only the inputs are observed and there are no target outputs. It is not a specific task-oriented rather indirect to the task at hand. Some of the Unsupervised learning approaches involve Principle Component Analysis (PCA), Independent Component Analysis (ICA), clustering, etc.

Many image classification approaches are based on the mentioned techniques [3][4][5]. These techniques infer the meaning from the semantic extraction of the image to determine the category in which it belongs. As a consequence, it can give expected classification for each image of different type like classifying between a human from animal. However, they can fail to accurately differentiate between similar images having comparable shape, and size.

1.1.2 **Supervised Learning**

In Supervised Learning, learning happens with a teacher. Here, the teacher represents a training dataset consisting of annotated labels to effectively train the algorithm. We know the correct answers, the algorithm iteratively makes predictions on the training data and is corrected by the teacher. Some of the Supervised learning techniques are Decision Trees, Neural Networks and more.

There are several approaches proposed with mentioned techniques [6][7]. These techniques involve learning from the features extracted from the images and label them to the target. In our experiment, we have considered neural networks to classify between images. Especially, convolutional neural networks (CNNs), due to superior accuracy through recognizing a pattern compared with classical machine learning methods, which rely on hand-crafted features. Also, many methods are formulated using CNNs [8][9] with high classification accuracy.

1.2 CONVOLUTIONAL NEURAL NETWORK

The convolutional neural network is a type of neural network which consists of synopsys/neurons that can be trained in terms of biases and weights. The image is processed in the form of a matrix for performing a convolution which is followed by a non-linear operation. The model works as a single differentiation function such that the raw image pixels are given as input and class scores were obtained at the end of the network.

It is known that CovNet models only take input in the form of images, which is useful to insert certain functions into the convolutional network. Consecutively makes these functional operations more efficient to implement and significantly reduce the number of variables in the network. A ConvNet model transforms the image into a 3D model (height, width, and depth) which gives a 3D output model from the neural network.

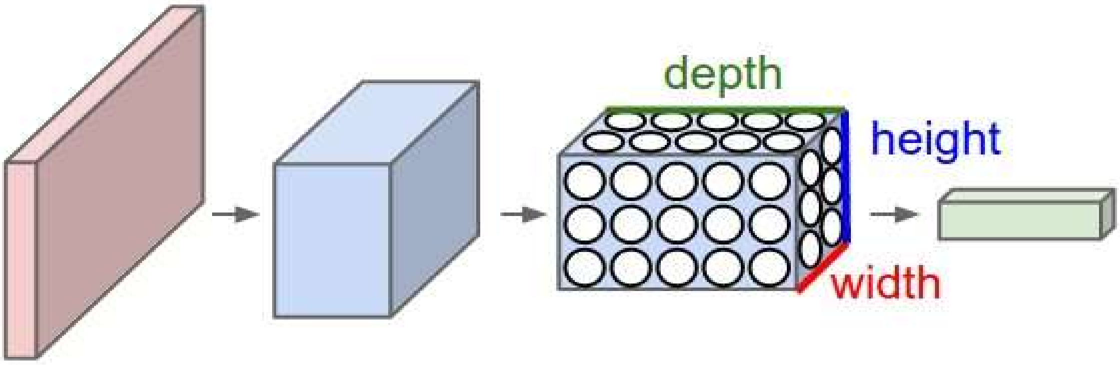


Figure 1. 1 Convolutional network

The CNN involves a series of tasks which performs the following operation.

1.2.1 **Convolution**

A Convolution involves moving a filter over the image matrix to extract the output feature map which gives new features. The value of the filter is predefined based on the type of feature you want to extract. However, it can be set according to the operation that you may desire. The size of the obtained output feature map is smaller as compared to the original image matrix.

While training, the CNN attains the ideal values for the feature mask after each iteration which helps it to extract features from the input image which are more meaningful. As the size of the feature map applied to the input increases, it proportionally increases the size of the output feature map obtained. However, the cutoff is with increasing the filter map size the computation involved for extracting output feature map also increases, hence it leads to more training time. Additionally, each filter added doesn't provide much change than the previous feature map, so we aim to construct neural networks that can extract the necessary feature for image classification with using the minimum number of filters. The convolution function can be written as.

$$h_j = f\left(h_{j-1} \otimes w_j + b_j\right) \tag{1.1}$$

Where $\otimes$ is the convolutional function and $f(x)$ is the activation function.

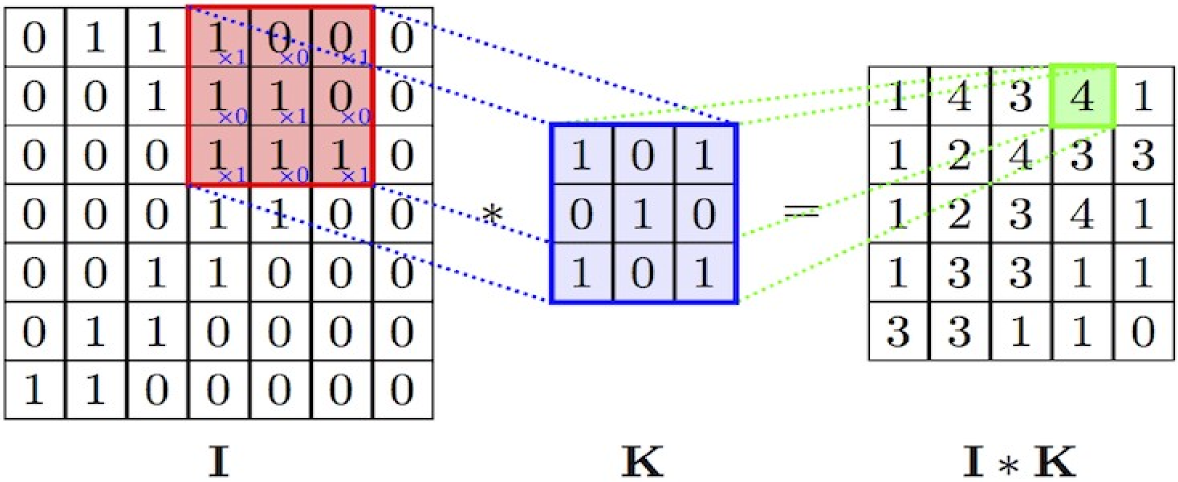


Figure 1. 2 Convolutional operation

1.2.2 **Relu**

After the convolution operation, the non-linearity is inserted on the data in order to tackle the overfitting problem. Furthermore, training the neural network for efficient image classification with different types of feature maps. There are a variety of non-linear functions are available, though Rectified Linear Unit which gives highly appropriate results in our model.

$$F(x) = \max(0, x) \tag{1.2}$$

returns x for all values of x > 0, and returns 0 for all values of x ≤ 0.

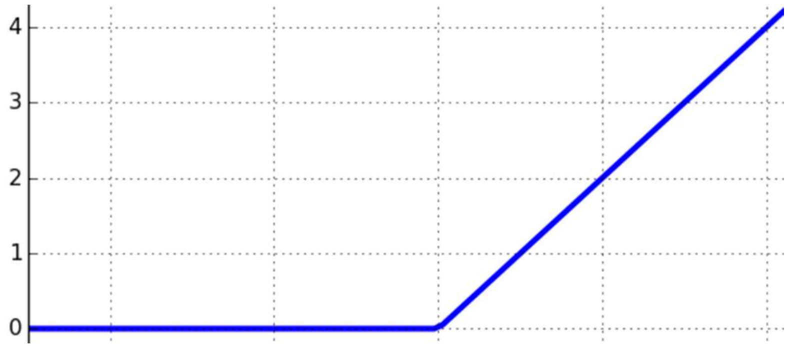


Figure 1. 3 ReLU activation function

1.2.3 Pooling

Following the ReLU a pooling step is performed, in which the CNN downsamples the extracted feature map, through reducing the number of dimensions of the feature map which sufficiently decreases the processing time, while still keeping the important features for further processing. There are many pooling methods available based in terms of sum, average, and max. In my model, I have used max pooling operation.

Max pooling is performed by sliding a window over the input feature map. For each feature map, the maximum value is considered while others are discarded.

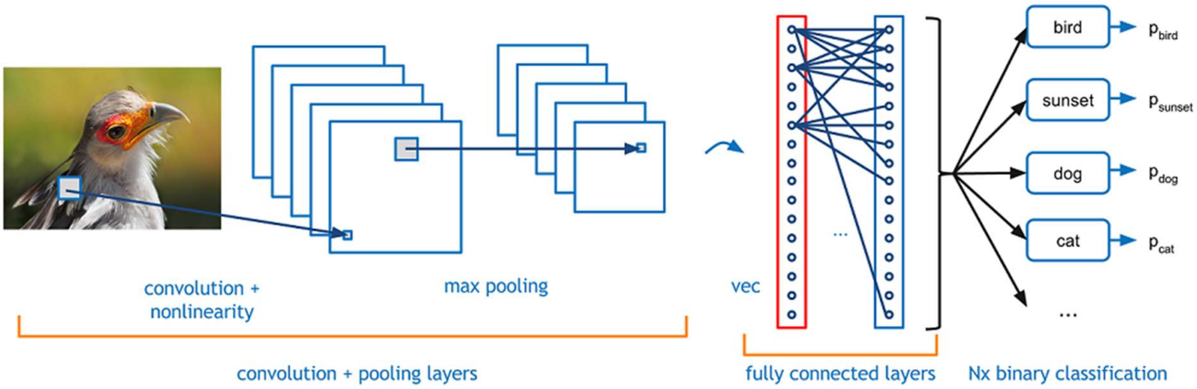


Figure 1. 4 An image classification model using ConvNet

1.3 PRE-TRAINED CONVOLUTIONAL NEURAL NETWORK

Pre-trained networks consist of trained weights according to a particular data set. If a model is built from scratch, then each synapse is initially assigned with random weights. However, after hours of training with a training dataset, it may converge in order to perform the desired task. After that, the accuracy can be measured through testing via blind dataset. Later those weights are saved for different problems which can reduce the overhead involved in training a model from scratch.

There are a variety of pre-trained neural networks available which were trained by large datasets. In our experiment, we have considered the VGG 16, VGG 19, and ResNet. Following we have explained their architecture and the type of weights through which they are initialized.

1.3.1 **VGG 16**

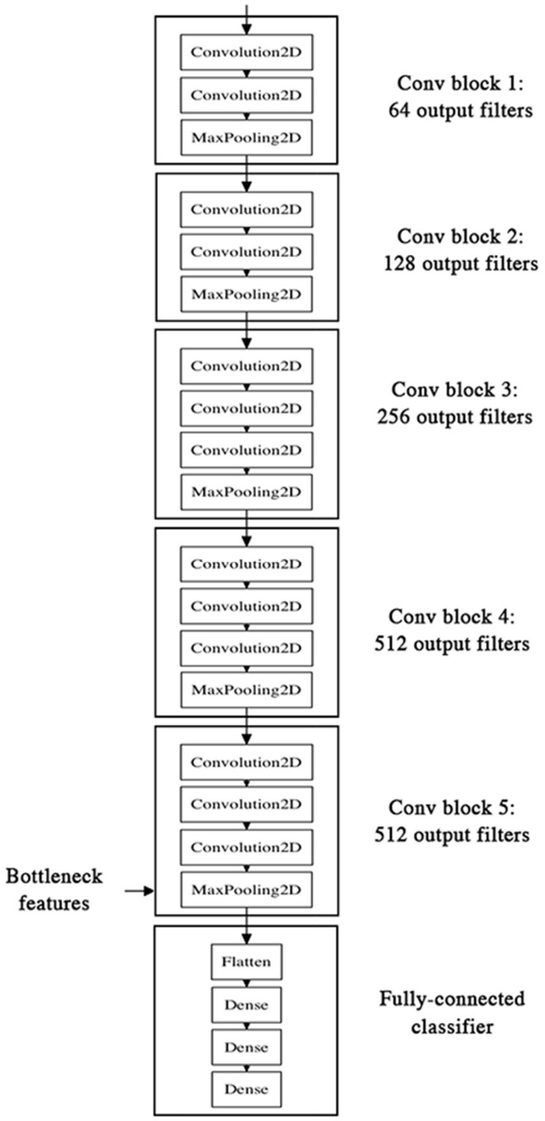
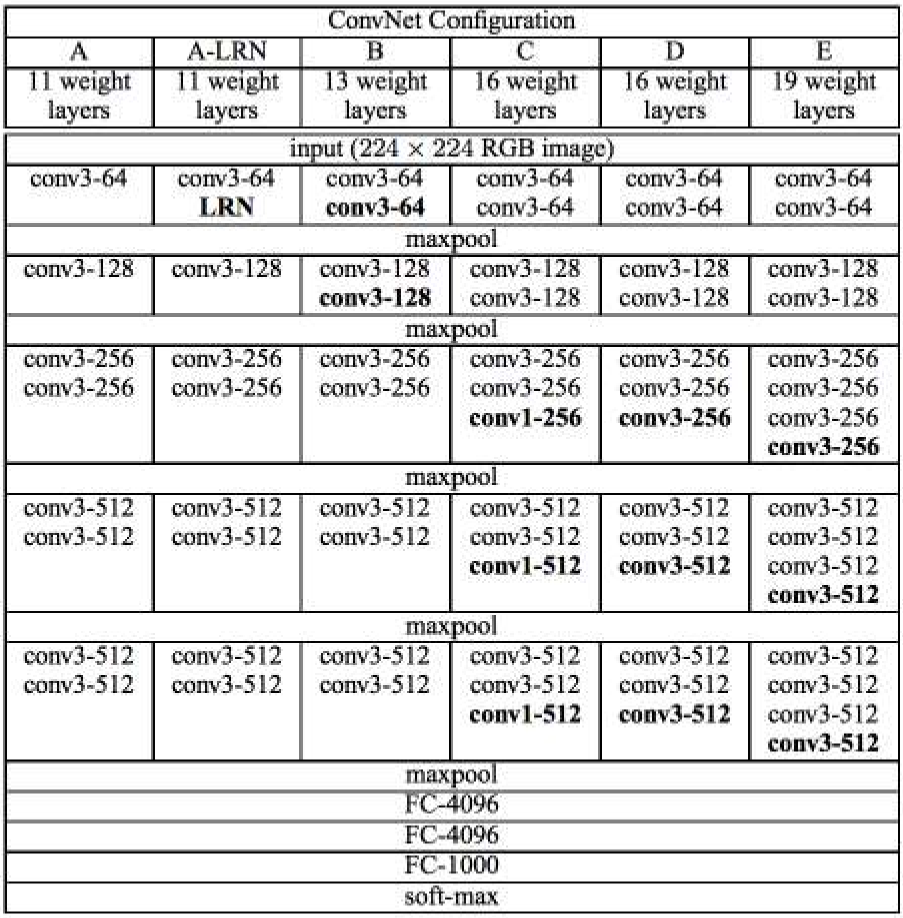


Figure 1. 5 VGG 16 architecture

VGG 16 network [10] is a pre-trained convolutional neural network trained on the ImageNet database. It consists of 5 blocks of the convolutional network each with max-pooling layers which are followed by a dense network, as visualized in figure 1.5. It has a total of 14 million trainable parameters with weights assigned as per the ImageNet database.

The ImageNet database has almost 15 million images of various subjects such as plants, animals, sport, artifact and many more. Due to its wide range of image data, the Vgg 16 network weights are useful in categorizing such subjects rapidly and accurately. It can achieve by transferring knowledge, called a transfer learning (discussed in section 1.4).

1.3.2 **VGG 19**



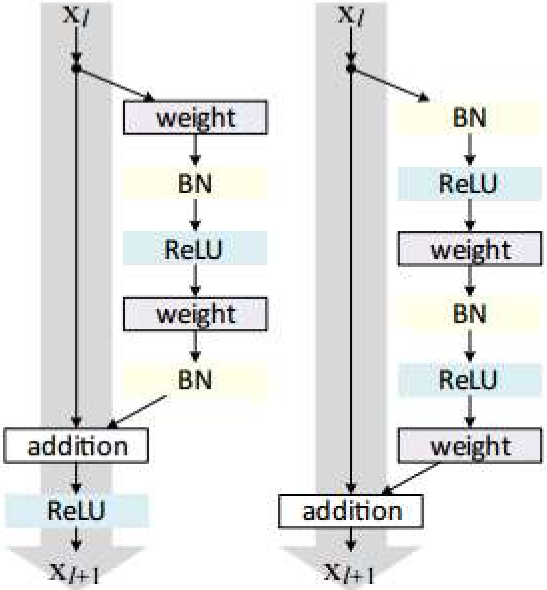
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 1. 6 VGG 19 architecture

VGG 19 model is similar to the VGG 16 model with an additional convolutional layer on block 3, 4 and 5. It has 21 million trainable parameters trained on ImageNet database. Although VGG net is useful for image classification problems, they're very slow to train due to the parameters of the network are very large. However, its depth and number of fully connected nodes of the VGG16, and VGG19 are over 533MB, and 574MB respectively.

1.3.3 **ResNet**

Apart from the well-known feed-forward network architectures such as AlexNet, OverFeat, and VGG. The ResNet [11] instead forms a network-in-network architecture which relies on micro-architecture modules.



(a) (b)

Figure 1. 7 (a) ResNet (b) ResNet using Identity mapping

The network-in-network architecture is formulated as residual functions. Through the experimental results, it has been observed that they can be optimized and obtain significant accuracy with increasing depth [12]. Even though ResNet is much deeper than VGG16 and VGG19, the size of the model is actually lesser than others due to the usage of global average pooling — which reduces the size of the model down to 102MB for ResNet50 (as in 50 weight layer).

1.4 TRANSFER LEARNING

Transfer learning refers to the process of using the weights of a pre-trained network trained on a large dataset applied to a different dataset (either as a feature extractor or by finetuning the network). Finetuning refers to the process of training the last few or more layers of the pretrained network on the new dataset to adjust the weight.

In our experiment, we have conducted both aspects of transfer learning in which we have taken the above discussed pre-trained networks which are either used as a feature extractor (known as bottleneck features) or finetuned the last few layers.

The framework of transfer learning is as follows in terms of domain, task, and marginal probabilities:

A domain, $\boldsymbol{D}$**,** is defined as a two-element tuple consisting of feature space , $\boldsymbol{\varkappa}$**,** and marginal probability, $\boldsymbol{P(X)}$, where $\boldsymbol{X}$ is a simple data point. Thus, we can represent the domain mathematically as $\boldsymbol{D} = \{\boldsymbol{\varkappa}, \boldsymbol{P(X)}\}$**.**

$$P(X), X = \{x_1, x_2, \dots\dots, x_n\}, x_i \in \varkappa \tag{1.3}$$

Here $\boldsymbol{x_i}$ represents a specific vector as represented in the above depiction. A task, $\boldsymbol{T}$**,** on the other hand, can be defined as a two-element tuple of the label space, $\boldsymbol{\gamma}$, and objective function, $\boldsymbol{\eta}$. The objective function can also be denoted as $\boldsymbol{P(\gamma/X)}$ from a probabilistic view point. For a given Domain $\boldsymbol{D}$**,** a Task is defined by two components.

$$T = \left\{\gamma, P\left(\frac{Y}{X}\right)\right\} = \{\gamma, \eta\} \qquad Y = \{y_1, \dots\dots, y_n\}, y_i \epsilon \gamma \tag{1.4}$$

γ = A label space

A predictive function η, learned from feature vector/label pairs, $(x_i, y_i), x_i \epsilon \varkappa, y_i \epsilon \gamma$.

For each feature vector in the domain, η predicts the corresponding label: $\eta(x_i) = y_i$.

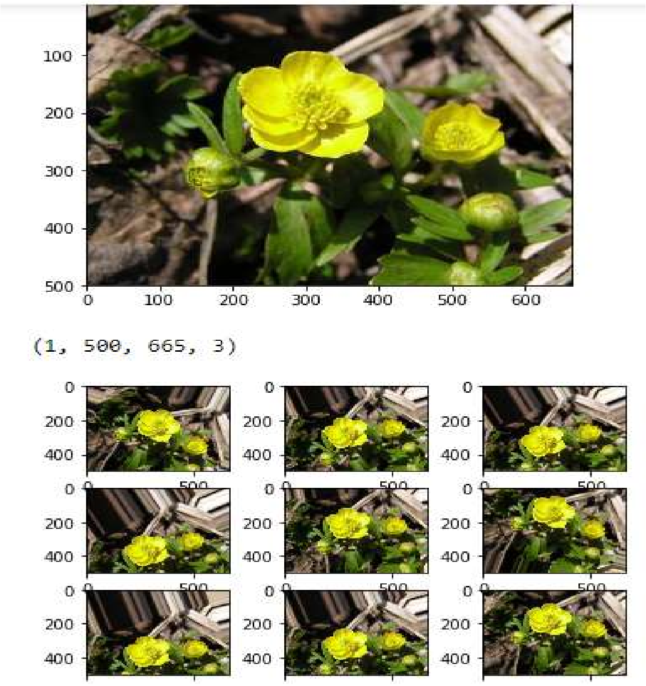
Thus, from the above definitions and representations, we can define transfer learning as follows.

Given a source domain $\boldsymbol{D}_s$, a corresponding source task $\boldsymbol{T}_s$, as well as a target domain $\boldsymbol{D_T}$ and a target task $\boldsymbol{T}_T$, the objective of transfer learning allow us to learn the target conditional probability distribution $\boldsymbol{P}(\boldsymbol{Y_T}|\boldsymbol{X_T})$ in $\boldsymbol{D}_T$ with the information gained from $\boldsymbol{D}_s$ and $\boldsymbol{T}_s$ where $\boldsymbol{D}_s \neq \boldsymbol{D_T}$ *or* $\boldsymbol{T_s} \neq \boldsymbol{T_T}$**.**

1.5 DATA AUGMENTATION

Usually, most of the popular datasets contain thousands of images, however, in a few rare cases fetching the right amount of data for our experiment became a tedious task. Furthermore, the data need to have a wide variety of objects in all dimensions, lighting angles, and structures in order to make our network model more versatile. However, to overcome the drawback of the limited amount of diverse data, we create our own dataset with the available data. This process of generating our own dataset is called as data augmentation [13].

Some of the data augmentation techniques which we have used in my experiment include rotation, scaling, translation, and perspective transform.



```
(1, 500, 665, 3)
```

Figure 1. 8 Performing data augmentation on a sample image

1.6 IMAGE SEGMENTATION

Image segmentation involves segregating the image into segments for further processing. With the wide variety of image segmentation techniques available, which partition the image into several parts based on certain image features like pixels, texture, intensity, etc. However, In our experiment, we have partitioned based on localizing the object of an image through a minimum bounding box.

There are many methods to do such localization, following give a brief description of a few.

1) U-Net
2) Fully connected network
3) Mask-RCNN

In our experiment, we have considered the architecture of the fully connected network.

1.6.1 U-Net

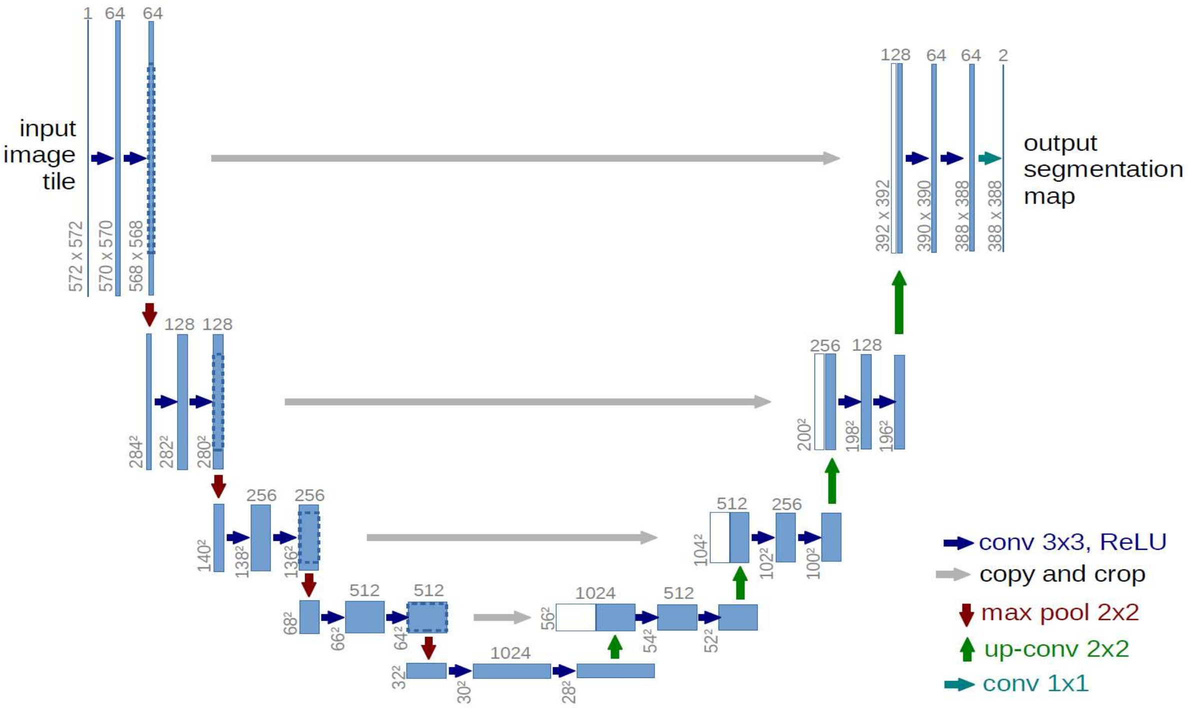


Figure 1. 9 U-Net architecture

U-Net [14] architecture is modeled in such a way that it could be trained in a few images and provide more segmentation accuracy. In this model, the pooling layer has been replaced with an upsampling layer which provides increased information to the upper layers.

Furthermore, the model doesn't have any fully connected layers which allow the model to segment the image through the overlapping tile strategy of the convolution operation. To overcome the deficiency of differentiating between with pixel classes of similar frequency weights of the filter map is computed as.

$$w(x) = w_c(x) + w_o.exp\left(-\frac{(d_1(x) + d_2(x))^2}{2\sigma^2}\right) \tag{1.5}$$

$x \in \Omega$, where x is the pixels in a image.

$\Omega \in Z^2$, also $w_c: \Omega \rightarrow R$.

1.6.2 Fully convolutional network

The fully connected network is an end-to-end connected network which performs semantic segmentation. However, it involves a skip connection for fine, deep image segmentation. The skip connection provides the feature map of a certain layer which is further used for processing in the next layer. Through, merging the feature map available at the variety of output layers to the corresponding output map gives a coarse segmented image with more detailed information.

In order to extract an efficient segmented map, the architecture of the Fully connected network is designed with a downsampling part followed by the upsampling part. Downsampling extracts the features from the images to interpret the data, whereas the upsampling process localizes the data.

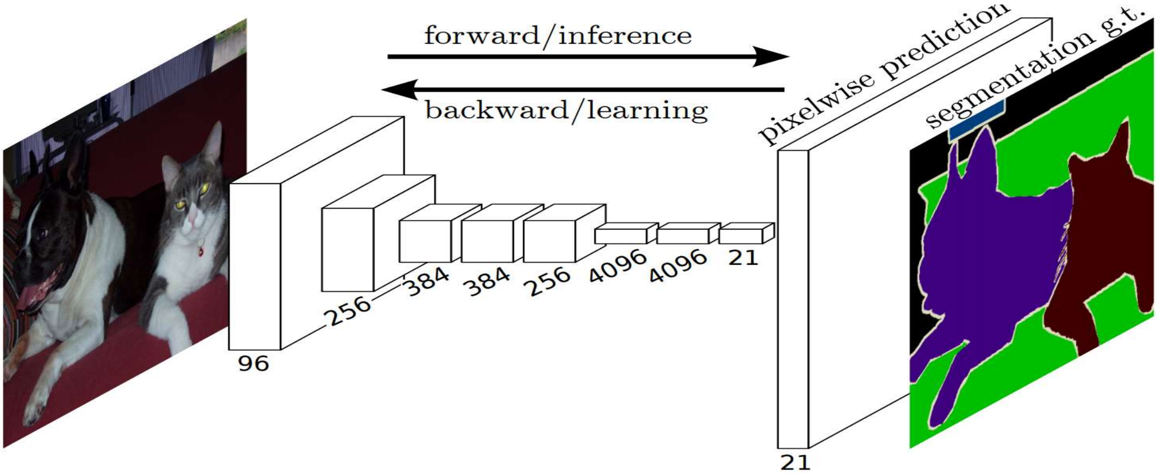


Figure 1. 10 Fully convolutional network

1.6.3 Mask – RCNN

Mask-RCNN [16] segments the images on the basis of the subjects presented on the image. The final output of Mask - RCNN involves creating a bounding box around the object and labeling it with a class. The architecture of the Mask – RCNN can be visualized as a combination of both Faster RCNN and FCN.

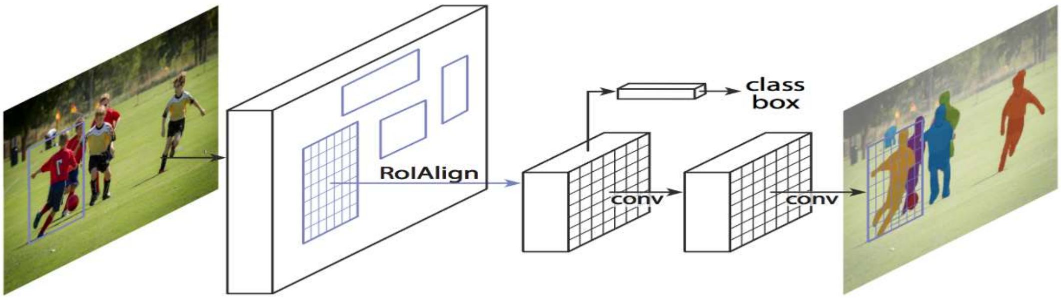


Figure 1. 11 The Mask – RCNN framework

1.7 BATCH NORMALIZATION

Batch normalization [17] reduces the time taken to train the neural network by decreasing the inherent covariant shift. The covariant shift occurs due to the parameters initialized at each layer were changed during training, as the parameter of the previous layer changes. This phenomenon causes the network to train slowly with careful parameter initialization and learning rate.

However, to increase the stability of the network while training it faster, batch normalization performs normalization operation on the output of a previous layer. It is performed through subtracting the input data with its mean and dividing by the standard deviation.

Apart from tuning the network to train faster, it can be useful for the following.

- Batch normalization works as a regularizer by adding some noise to the data. Therefore, it limits the usage of the dropout layer through which some essential information could be lost.
- It also provides non-linearity in the data which causes the network to train more independently of other layers.

However, batch normalization is not initialized in the pre-trained network which can hamper the learned weights on the network. Although, I have used it in the fully connected network for efficient training of the network.

Input: values of x over a mini batch: $B = \{x_{1....m}\}$

$$\boldsymbol{\mu_B} = \frac{\mathbf{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}} \boldsymbol{x_i} \tag{1.6}$$

$$\boldsymbol{\sigma_B^2} = \frac{\mathbf{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}}\left(\boldsymbol{x_i} - \boldsymbol{\mu_B}\right)^{\mathbf{2}} \tag{1.7}$$

$$\widehat{\boldsymbol{x_\imath}} = \frac{\boldsymbol{x_i - \mu_B}}{\sqrt{\boldsymbol{\sigma_B^2} + \in}} \tag{1.8}$$

CHAPTER 2 RELATED WORK

In this section, we will present related work which addresses the image classification and segmentation task. Also, the recent work on dog breed and bird species classification through deep learning and non-deep learning techniques.

2.1 IMAGE SEGMENTATION

There are a variety of ways has been proposed for image segmentation which can perform binary segmentation: 0 for background and 1 for the object region(s). One of which is YOLO [18] (You only look once) algorithm, for object classification and localization using bounding boxes. It consists of 24 layers of single CNN network which is followed by a dense network at the end. However, to reduce the training period we can opt for fast YOLO (9 layers of single CNN network instead of 24).

Though the data availability, there may be a possibility of fewer training samples to train the model effectively for segmentation. U-Net is one of the methods to train with fewer annotated labels, but they can only process 2D images. V-Net [19] can perform 3D segmentation based on volumetric, fully convolutional, neural network. Furthermore, various other methods are proposed which uses CNN to do the object localization process such as Mask-RCNN. Overall, all of the mentioned methods use the concept of the fully convolutional network (FCN) for semantic segmentation.

2.2 DOG BREED AND BIRD SPECIES IMAGE CLASSIFICATION

Various methods have been proposed to classify dog breed and bird species images. The greater number of methodologies have used machine learning based methods. For instance, the work in [20] which coarsely classify dog breeds in 5 groups and for each group principal component analysis (PCA) is used as a feature extractor. The extracted features are compared with the feature template which was derived while classifying the dog images into groups. The image is classified as the breed that gives the minimum distance between two feature vectors.

In [21], they model the classification of a dog as a manifold which uses facial shapes and their associated geometry. They have applied Grassmann manifold to represent the geometry of different dog breeds. After feature extraction, SVM categorizes the features to the corresponding category. An improved dog breed classification is proposed [22] from click-through logs and pre-trained models. In this approach, more dog images are mined through the web with dog-related keywords for gathering more information. After that, important information is derived using pre-trained model (DCNN) by keeping related neurons in the last layers. Later, those features are classified through a dense neural network for accurate classification.

To classify among different bird species several methods have been proposed, in [23] the authors have extracted textural content of spectrogram images. These features are extracted through three different techniques of digital image processing: Gabor features, Local Binary Pattern (LBP) and Local Phase Quantization (LPQ). After that, the SVM classifier is used to classify these textural content and the results were taken as 10 fold-cross validation.

Though the basic feature extraction methods can provide valuable information for categorizing the images. Although, it cannot achieve higher classification accuracy in comparison with deep learning techniques. However, hybrid of techniques were used to classify between bird species as shown in work [24] using decision tree and SVM classifier. In addition, the classification of bird species is performed through bird voices [25] using CNN.

CHAPTER 3 THE EXPERIMENTAL APPROACH

The following flow diagram illustrates our approach for classifying between two different datasets of dog breeds and bird species.

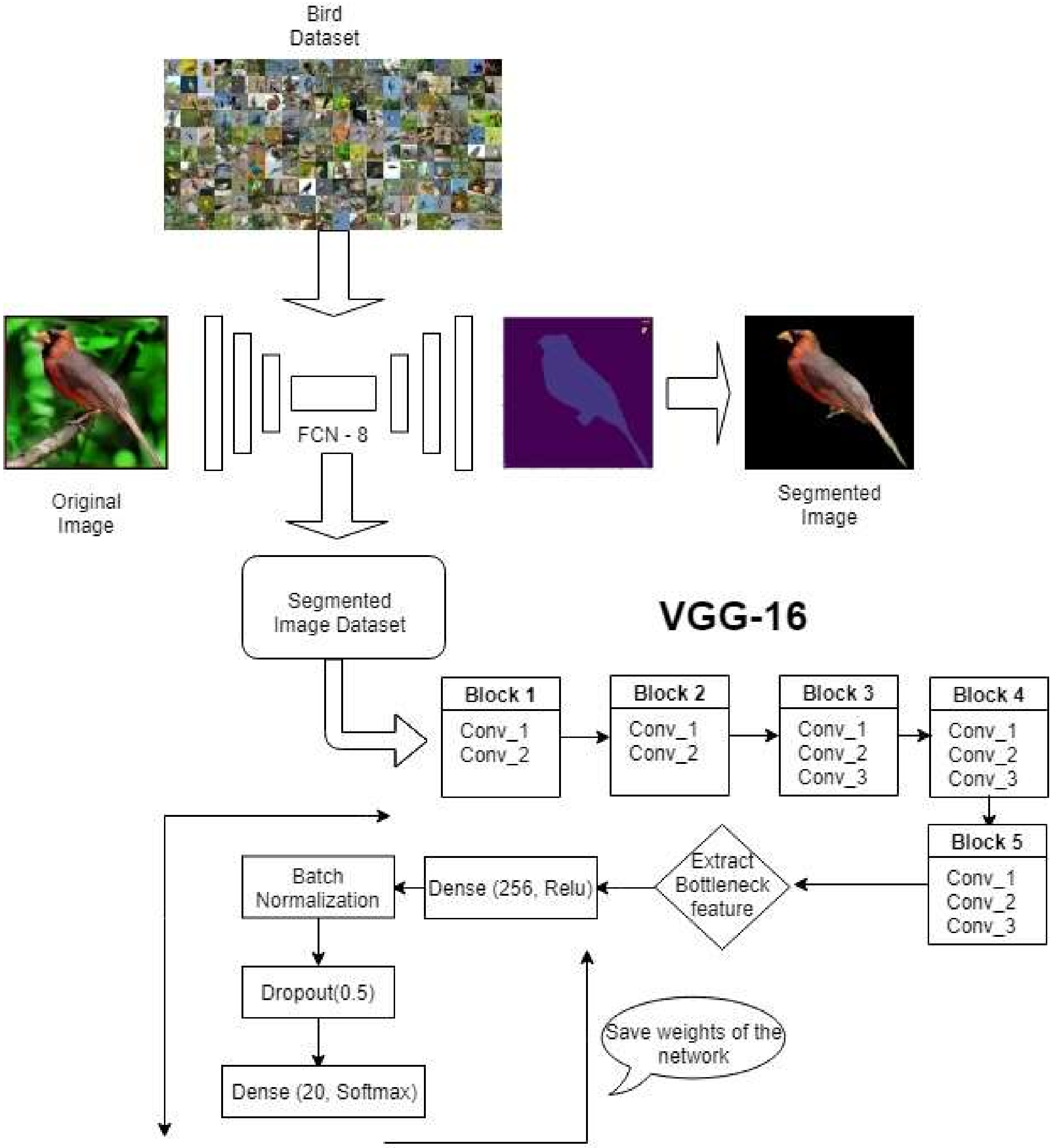


Figure 3. 1 First phase of bird species classification

We have conducted a two-phase experiment which follows the flow as shown in figure 3.1 and 3.3 for bird species classification. The first phase consists of the whole layers of

VGG 16 to extract features from the segmented images. Here the VGG 16 model performs as feature extractor with ImageNet weights. Later, we have used those bottleneck features to train the dense network. Batch normalization is performed on the output of the first layer of the dense network to normalize the data such that the mean of the data becomes 0 and the standard deviation equals to 1. Furthermore, those extracted features are used to train the dense network for image classification until it converges or it doesn't improve for five epochs. The best weights associated with the dense network is saved for fine-tuning purposes on the second phase of the experiment.

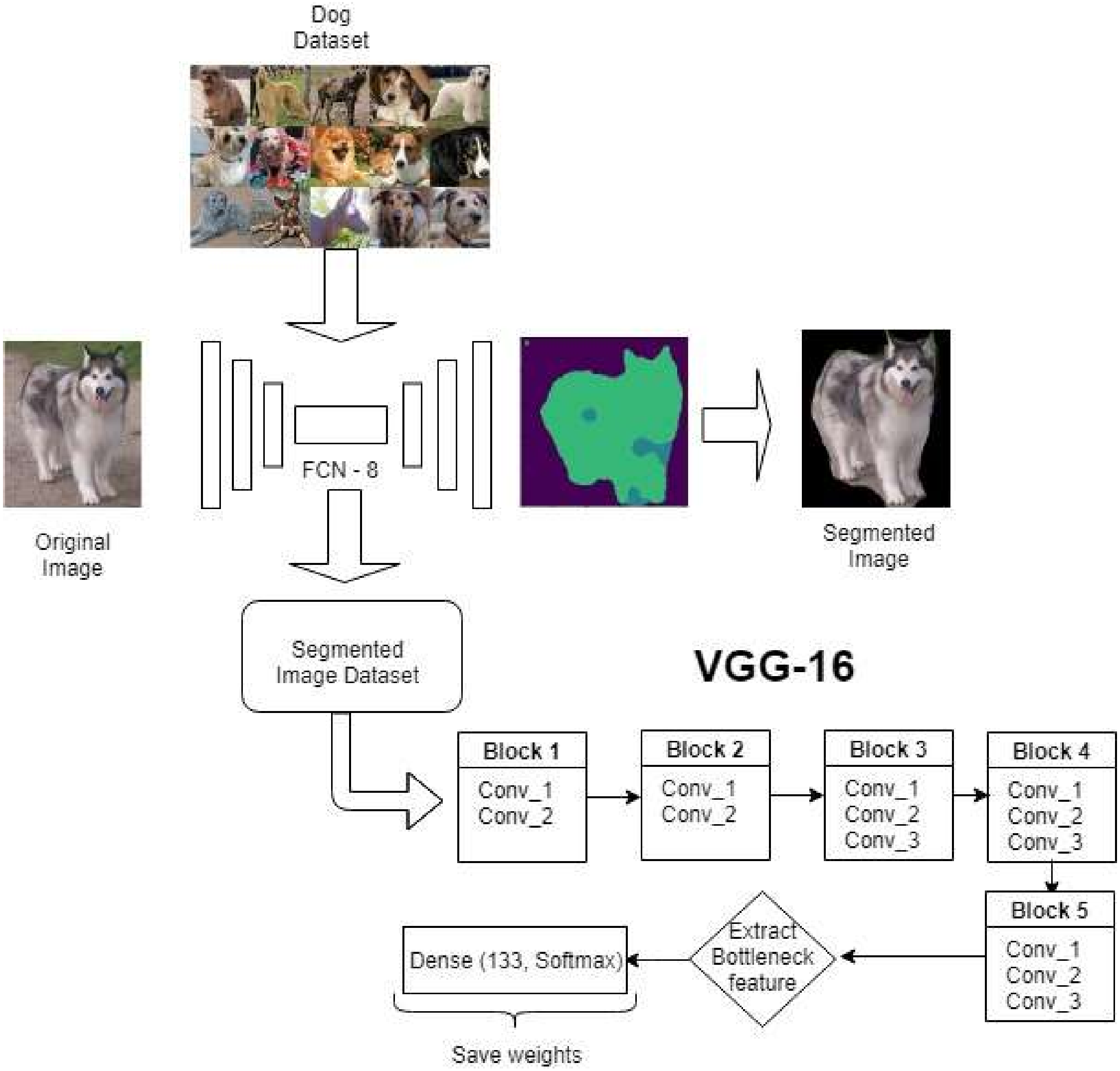


Figure 3. 2 First Phase of dog breed classification

In the second phase, the dense layer is updated with the best-saved weights from the first phase of the experiment. Also, the last layer of the VGG 16 network remained trainable while the other layers were frozen. Finally, we have trained the network once again by rescaling the image through data augmentation.

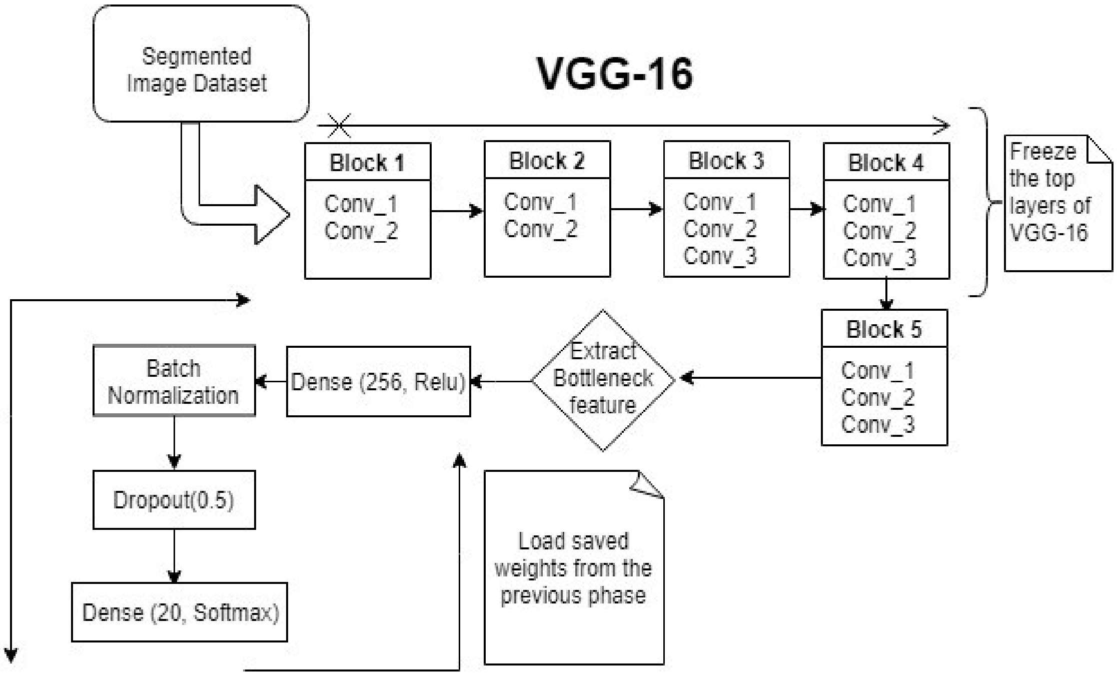


Figure 3. 3 Second phase of bird species classification

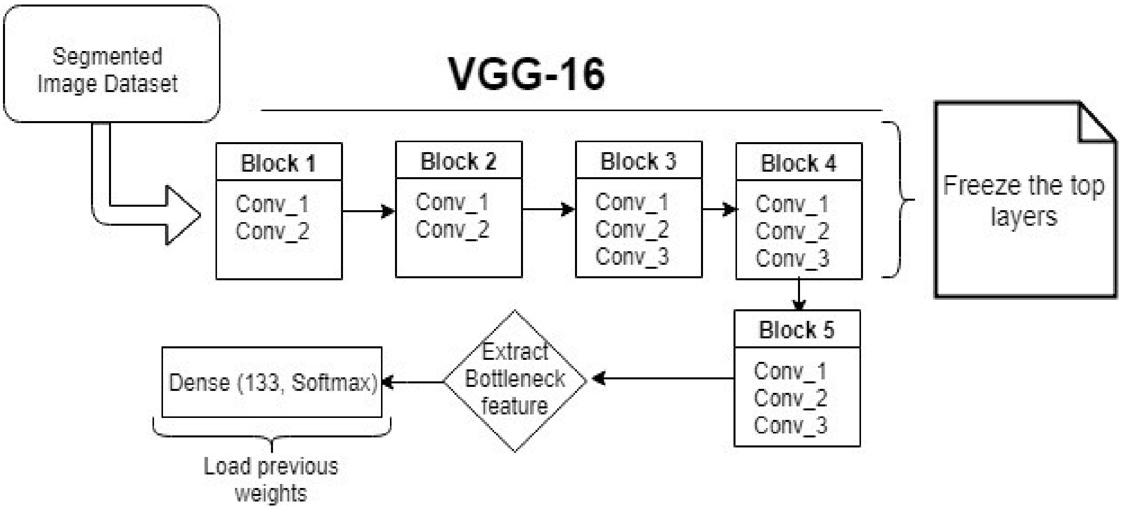


Figure 3. 4 Second phase dog breed classification

3.1 FCN FOR SEMANTIC IMAGE SEGMENTATION

Those image datasets which we have considered consists of a lot of background subjects like a person, plant, and shapes. It became a challenging task to get ideal features of the subject, which makes the problem of automatic image classification a cumbersome task.

Therefore, we propose an automatic image segmentation process with the fully convolutional network (FCN) by segmenting only through the region of the object. We have formulated the segmentation task as binary classification process: 0 for background and 1 for the object region(s).

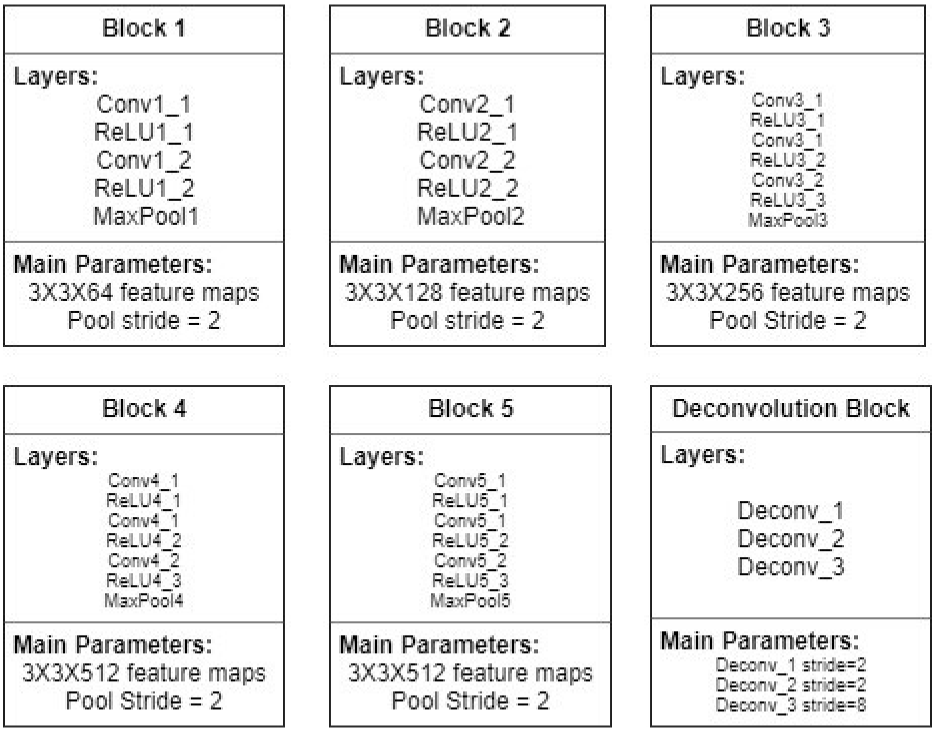


Figure 3. 5 FCN-8 architecture

Initially, a mask is created from the FCN network which concentrates on the subject region. Later that mask is used to extract that subject while making the background region black for removing the noise.

The fully connected network is translation invariant – it can segment the image even it is subjected to any type of transformation as it depends on the relative spatial coordinates. $x_{i,j}$ represents the data vector at the location *(i, j)* in a particular layer, and y_{ij} for the following layer, the following function compute outputs y_{ij} through

$$y_{ij} = f_{ks}(\{x_{si} + \delta i, sj + \delta j\}0 \leq \delta i, \delta j \leq k) \tag{3.1}$$

Where s is the stride or subsampling factor, k is called the kernel size, and f_{ks} determines the layer type: convolution is depicted by a matrix multiplication, taking out the max element in max pooling, or performing a non-linear operation on each element from activation function which is continued for other layers.

This functional form is calculated, with kernel size and stride satisfying the transformation rule.

$$f_{ks} o g_{\acute{k}\acute{s}} = (f \ o \ g)_{\acute{k}} + (k-1)_{\acute{s},s\acute{s}} \tag{3.2}$$

Per-pixel accuracy,

$$acc(P, GT) = \frac{|pixels\ correctly\ predicted|}{|total\ number\ of\ pixels|} \tag{3.3}$$

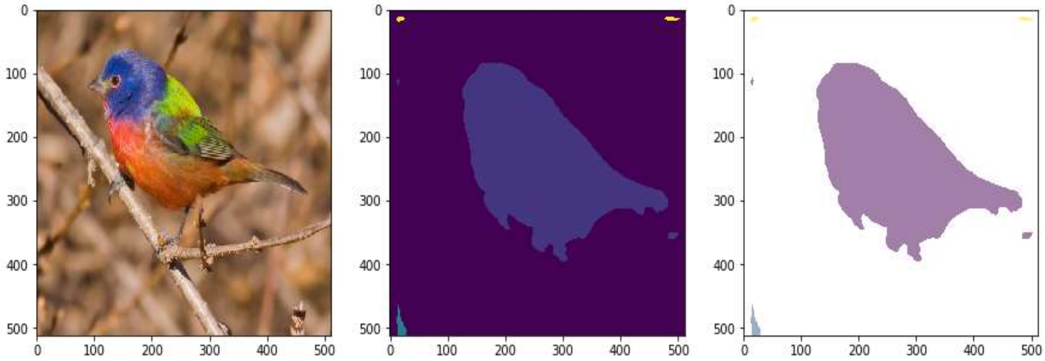


Figure 3. 6 Mask obtained through FCN-8 network

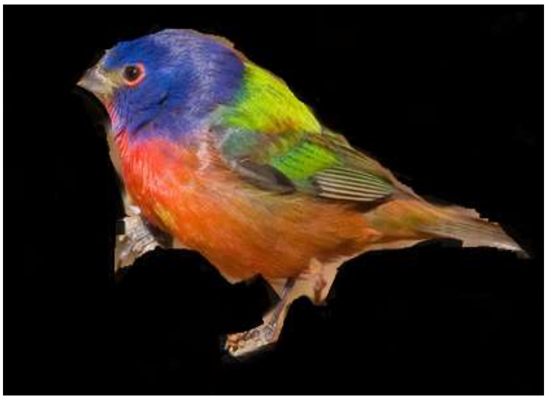
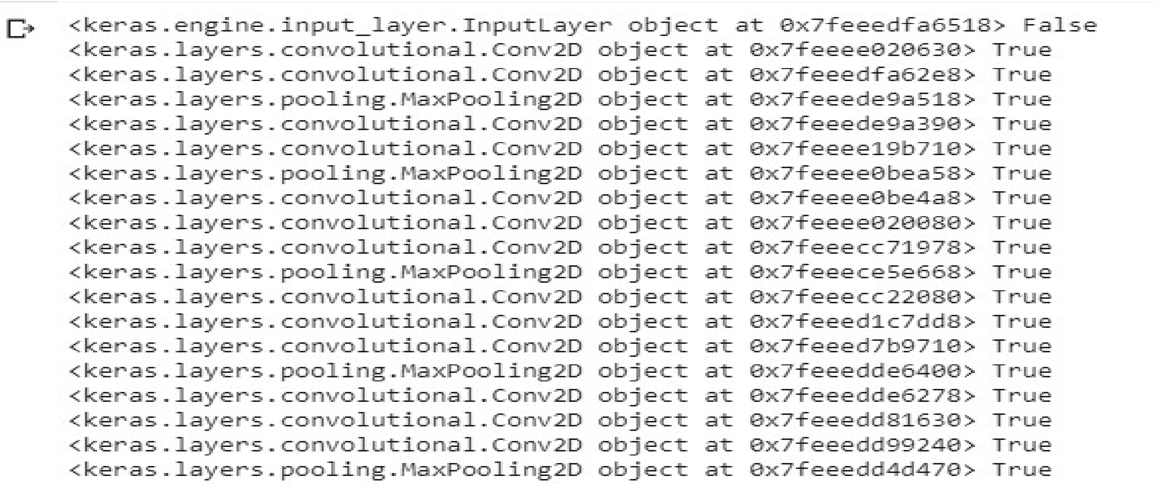


Figure 3. 7 Segmented images after masking

3.2 VGG 16 MODEL AS A FEATURE EXTRACTOR

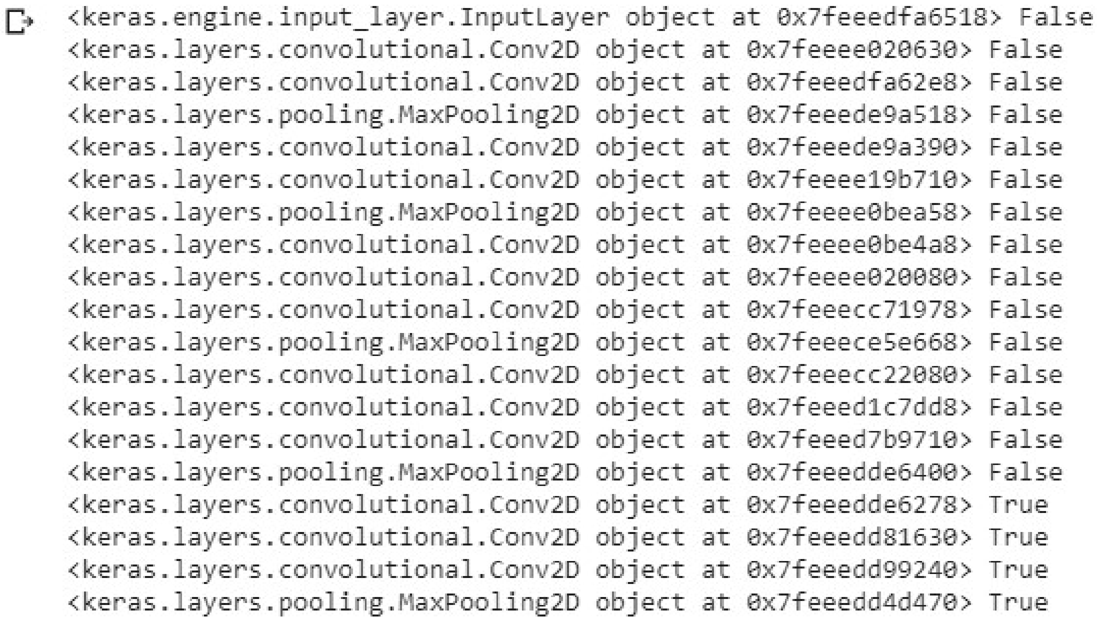
After segmenting the images, highly discriminative regions remained while other possible misleading regions are removed. In this work, we have passed those segmented images through the VGG 16 model which performs as a feature extractor initialized with the ImageNet weights. Each block in the VGG 16 model is a network of convolutional layers which is followed by a pooling layer to reduce the feature map. Currently, most of the methods extract features of an image by using every layer of VGG 16 model. However, we can freeze some layers to extract features from the remaining available layers.

In our work, the VGG 16 model is used as a feature extractor in first-phase then those features are used to train the dense network. Although, in the second phase we have frozen the top layers and trained in conjunction with the dense network.



```
<keras.engine.input_layer.InputLayer object at 0x7feeedfa6518> False
<keras.layers.convolutional.Conv2D object at 0x7feeee020630> True
<keras.layers.convolutional.Conv2D object at 0x7feeedfa62e8> True
<keras.layers.pooling.MaxPooling2D object at 0x7feeede9a518> True
<keras.layers.convolutional.Conv2D object at 0x7feeede9a390> True
<keras.layers.convolutional.Conv2D object at 0x7feeee19b710> True
<keras.layers.pooling.MaxPooling2D object at 0x7feeee0bea58> True
<keras.layers.convolutional.Conv2D object at 0x7feeee0be4a8> True
<keras.layers.convolutional.Conv2D object at 0x7feeee020080> True
<keras.layers.convolutional.Conv2D object at 0x7feeecc71978> True
<keras.layers.pooling.MaxPooling2D object at 0x7feeece5e668> True
<keras.layers.convolutional.Conv2D object at 0x7feeecc22080> True
<keras.layers.convolutional.Conv2D object at 0x7feeed1c7dd8> True
<keras.layers.convolutional.Conv2D object at 0x7feeed7b9710> True
<keras.layers.pooling.MaxPooling2D object at 0x7feeedde6400> True
<keras.layers.convolutional.Conv2D object at 0x7feeedde6278> True
<keras.layers.convolutional.Conv2D object at 0x7feeedd81630> True
<keras.layers.convolutional.Conv2D object at 0x7feeedd99240> True
<keras.layers.pooling.MaxPooling2D object at 0x7feeedd4d470> True
```

Figure 3. 8 VGG 16 model with all trainable layers in the first phase

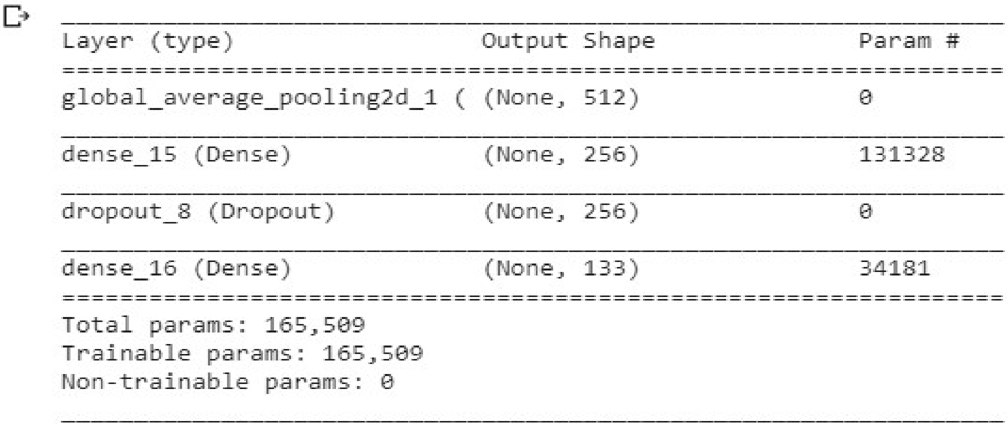


```
<keras.engine.input_layer.InputLayer object at 0x7feeedfa6518> False
<keras.layers.convolutional.Conv2D object at 0x7feeee020630> False
<keras.layers.convolutional.Conv2D object at 0x7feeedfa62e8> False
<keras.layers.pooling.MaxPooling2D object at 0x7feeede9a518> False
<keras.layers.convolutional.Conv2D object at 0x7feeede9a390> False
<keras.layers.convolutional.Conv2D object at 0x7feeee19b710> False
<keras.layers.pooling.MaxPooling2D object at 0x7feeee0bea58> False
<keras.layers.convolutional.Conv2D object at 0x7feeee0be4a8> False
<keras.layers.convolutional.Conv2D object at 0x7feeee020080> False
<keras.layers.convolutional.Conv2D object at 0x7feeecc71978> False
<keras.layers.pooling.MaxPooling2D object at 0x7feeece5e668> False
<keras.layers.convolutional.Conv2D object at 0x7feeecc22080> False
<keras.layers.convolutional.Conv2D object at 0x7feeed1c7dd8> False
<keras.layers.convolutional.Conv2D object at 0x7feeed7b9710> False
<keras.layers.pooling.MaxPooling2D object at 0x7feeedde6400> False
<keras.layers.convolutional.Conv2D object at 0x7feeedde6278> True
<keras.layers.convolutional.Conv2D object at 0x7feeedd81630> True
<keras.layers.convolutional.Conv2D object at 0x7feeedd99240> True
<keras.layers.pooling.MaxPooling2D object at 0x7feeedd4d470> True
```

Figure 3. 9 VGG 16 model with only last trainable layer in the second phase

3.3 DENSE NETWORK

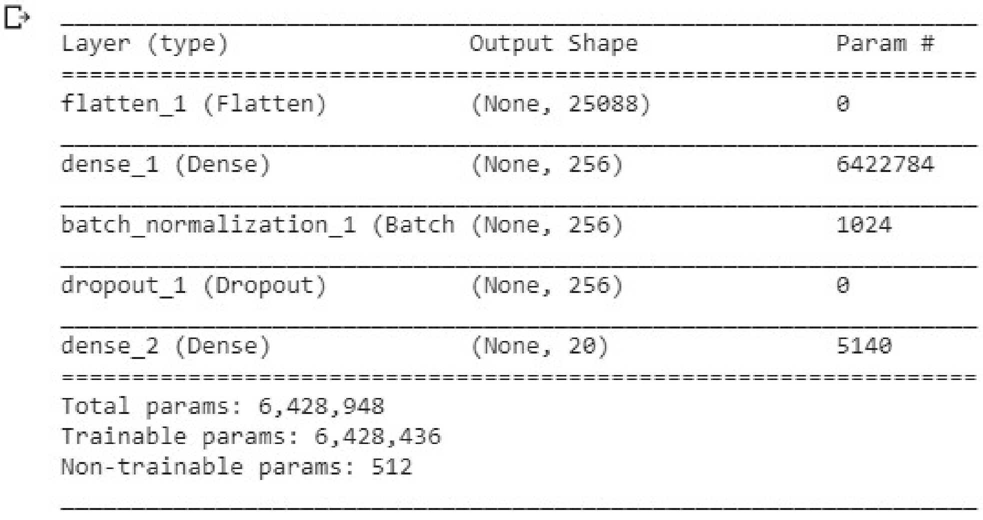
In both phases of the experiment, the dense network architecture remains the same because saved weights can be a load if it has the same architecture. We have built two different dense layers for each dataset separately (dog breed and bird species).



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
global_average_pooling2d_1 ( (None, 512)               0
_________________________________________________________________
dense_15 (Dense)             (None, 256)               131328
_________________________________________________________________
dropout_8 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_16 (Dense)             (None, 133)               34181
=================================================================
Total params: 165,509
Trainable params: 165,509
Non-trainable params: 0
_________________________________________________________________
```

Figure 3. 10 Dense networks for Dog breed classification

For dog breed classification, the first dense layer has 256 output shape with Relu activation function to cause nonlinearity between features. It is densely connected to another layer with 133 output shape with softmax function which causes the output to be in probabilistic form. Between these connections of dense layers there a dropout layer with a probability of 30% which states that 30% of the connections will be dropped during execution. This is performed to prevent the overfitting of data and perform regularization.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten_1 (Flatten)          (None, 25088)             0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               6422784
_________________________________________________________________
batch_normalization_1 (Batch (None, 256)               1024
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 20)                5140
=================================================================
Total params: 6,428,948
Trainable params: 6,428,436
Non-trainable params: 512
_________________________________________________________________
```

Figure 3. 11 Dense networks for Bird species classification

For bird species classification, the initial dense layer has 256 output shape with Relu activation function to cause nonlinearity between features. Then batch normalization is performed at the end of dense layers to normalize the features such that mean is 0 and standard deviation equal to 1. Further, it is connected to another dense layer with 20 output shape to categorize between 20 different bird species with a softmax activation function. Between the connections of these dense layers, there is a dropout layer with 20% probability.

3.4 DATA AUGMENTATION FOR THE TRAINING SET

In our experiment, we have performed data augmentation on the training set for efficient training of the model. We have rescaled each image between 0 or 1 for further processing of the image. Since each image consists of RGB coefficients ranging from 0 to 255, but it would be difficult to process the image. So, instead, we have targeted to rescale the images between 0 and 1 instead of by scaling with a 1./255 factor.

CHAPTER 4 EXPERIMENTAL RESULTS

The results are measured in terms of its accuracy to classify the images as per the categorized labels. The dataset used for the evaluation process was Caltech-USCD Birds 200 for bird image classification and Stanford dataset for dog classification.

The dataset used for bird classification consists of images of 200 different types of bird species with a total of 6033 images. We have divided the dataset into three different datasets which are train set which is used to train our model, validation set for validating our results and test set predict the image on sample images. Initially, we have conducted the operation on 20 different bird species, where we have a total of 1115 images in which train set consist of 789 images while validation set contains 267 images and 59 images in the test set. However, for each type of species, there are around 49 train images which are very much less number of images to train our model efficiently.

Figure 4. 1 Bird dataset

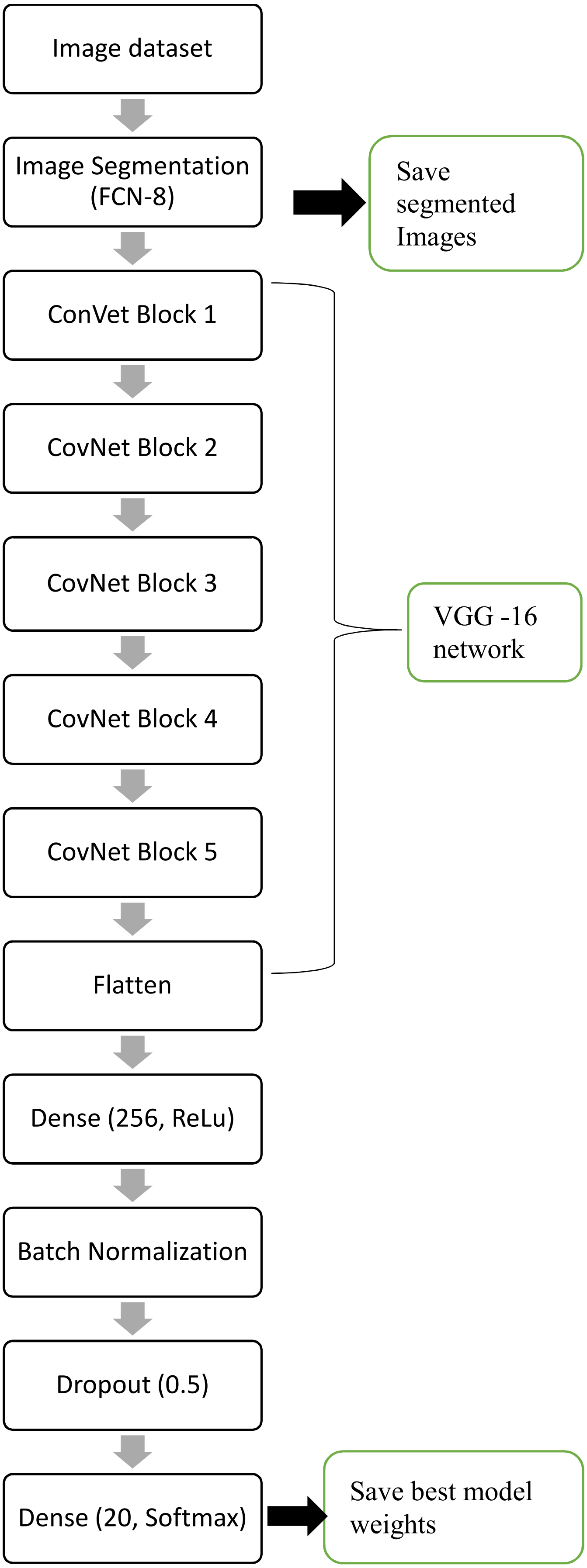


Figure 4. 2 Phase -1 of the proposed algorithm

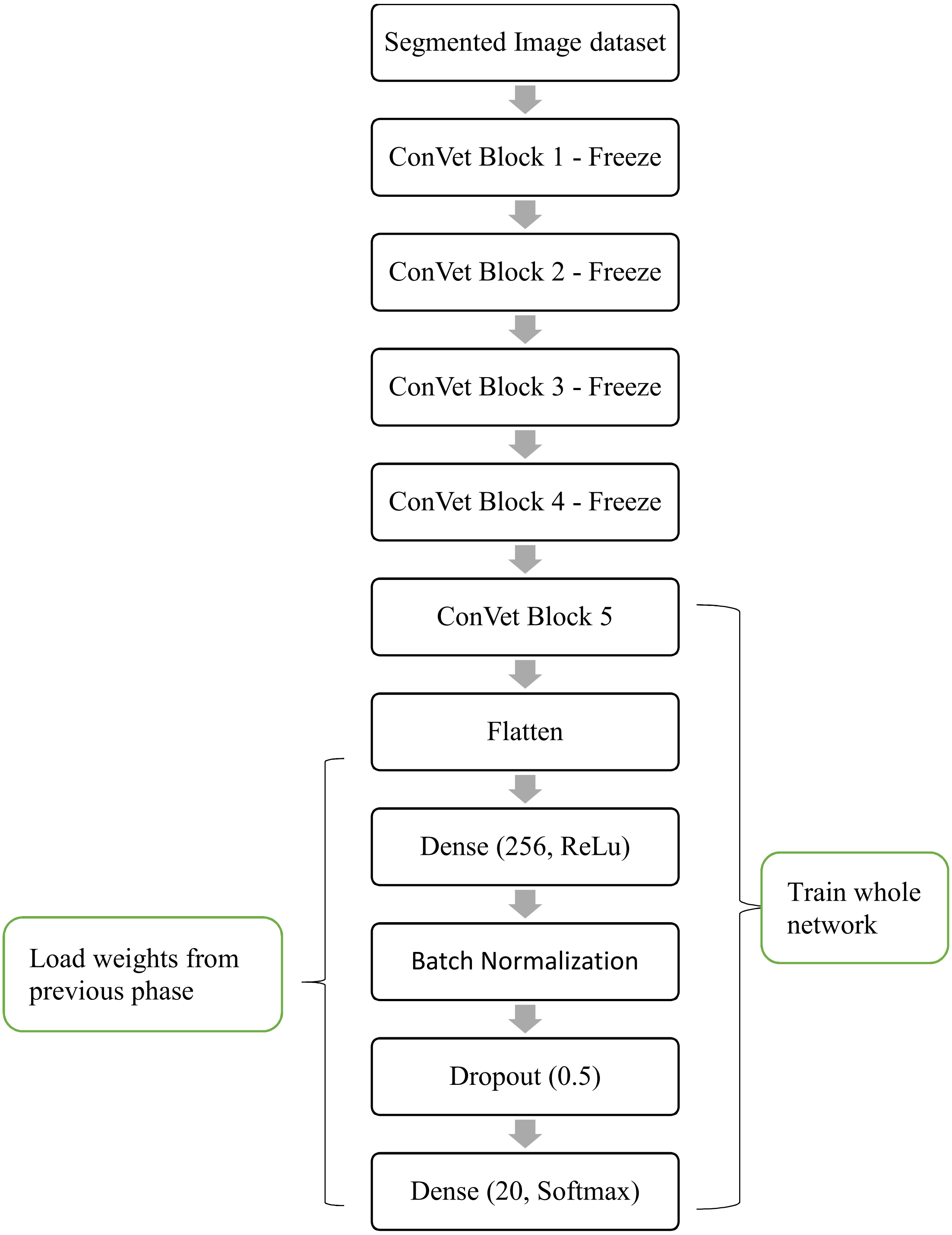


Figure 4. 3 Phase-2 of the proposed algorithm

We have performed the segmentation process through FCN-8 for extracting the subject region while removing the background area which consists of noise. The segmented dataset is passed to phase one of my proposed algorithm which extracts the best model weights for the dense network by training through the features extracted from the VGG-16 network. Later those saved model weights are loaded to the dense network of my second phase of the algorithm for fine-tuning it with the VGG-16 network. Finally, we have used the fine-tuned trained network for classifying between various bird species.

The classification accuracy achieved through my proposed algorithm is 97% which exceeds the other algorithms as mentioned in table 4.1. We have compared my algorithm's accuracy if the segmentation process is not performed to validate the importance of removing the background region.

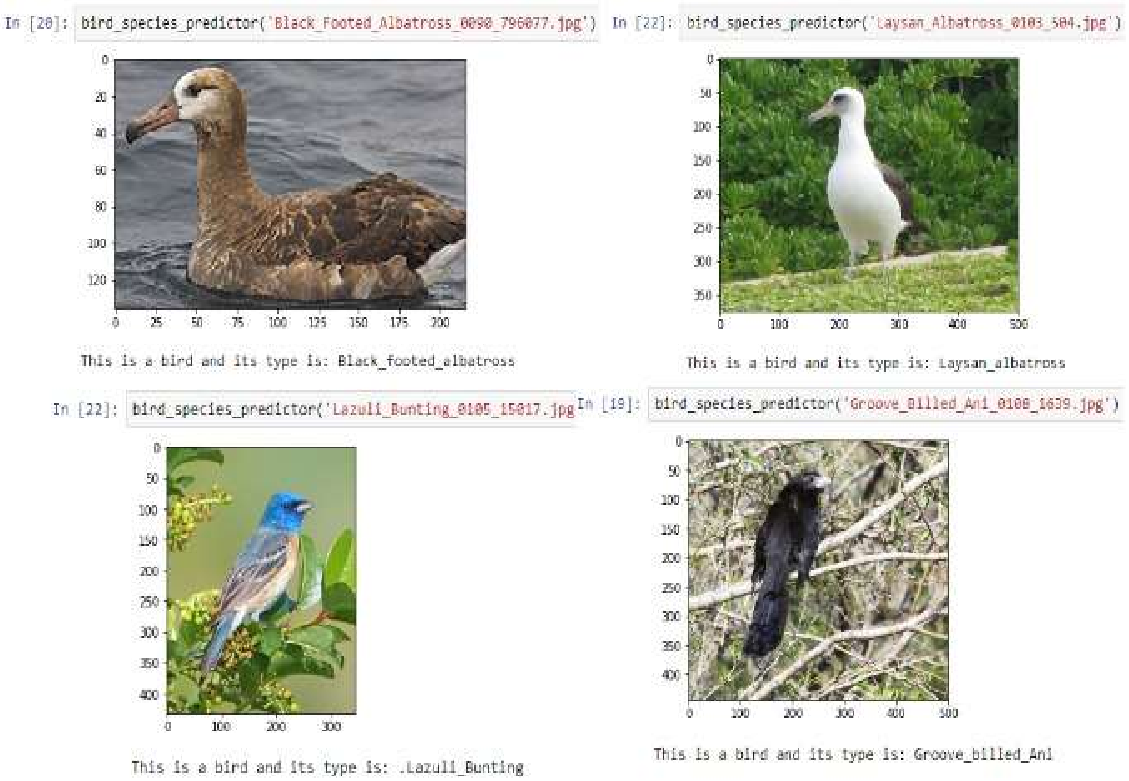
$$Accuracy = \frac{|Number\ of\ images\ correctly\ classified|}{|Total\ number\ of\ images\ in\ test\ set|} \quad (4.1)$$

S. NO.	Method	Image Segmentation (Y/N)	Accuracy (%)
1.	VGG -16 model with a two-layer dense network	N	49.23
2.	VGG -16 model with a two-layer dense network	Y	54.62
3.	VGG -19 model with a two-layer dense network	N	44.43
4.	VGG -19 model with a two-layer dense network	Y	46.15
5.	Proposed Two-phase method	Y	94.89

Table 4. 1Bird classification

S. NO.	Method	Batch Normalization (Y/N)	Accuracy (%)
1.	VGG -16 model with a two-layer dense network	N	50.00
2.	VGG -19 model with a two-layer dense network	N	42.31
3.	Proposed Two-phase method	N	89.84

Table 4. 2The difference using Batch normalization in Bird classification



```
In [20]: bird_species_predictor('Black_Footed_Albatross_0090_796077.jpg')
```

```
This is a bird and its type is: Black_footed_albatross
```

```
In [22]: bird_species_predictor('Laysan_Albatross_0103_504.jpg')
```

```
This is a bird and its type is: Laysan_albatross
```

```
In [22]: bird_species_predictor('Lazuli_Bunting_0105_15017.jpg
```

```
This is a bird and its type is: .Lazuli_Bunting
```

```
In [19]: bird_species_predictor('Groove_Billed_Ani_0100_1639.jpg')
```

```
This is a bird and its type is: Groove_billed_Ani
```

Figure 4. 4 Examples of correctly classified images

The data set used for dog classification consists of 133 different dog breeds of total 8351 images. The train set consist of 6680 images while the validation set consist of 835 images and test set contains 836 images. Here, we will use the similar VGG-16 model methodology to fine-tune with the fully connected dense network.

We have performed a similar process of bird species classification for dog breed classification with a slight difference of dense network for higher accurate differentiation. The classification accuracy achieved through my proposed algorithm is 98% which exceeds the other algorithms as mentioned in table 4.2. We have compared my algorithm's accuracy if the segmentation process is not performed to validate the importance of removing the background region.

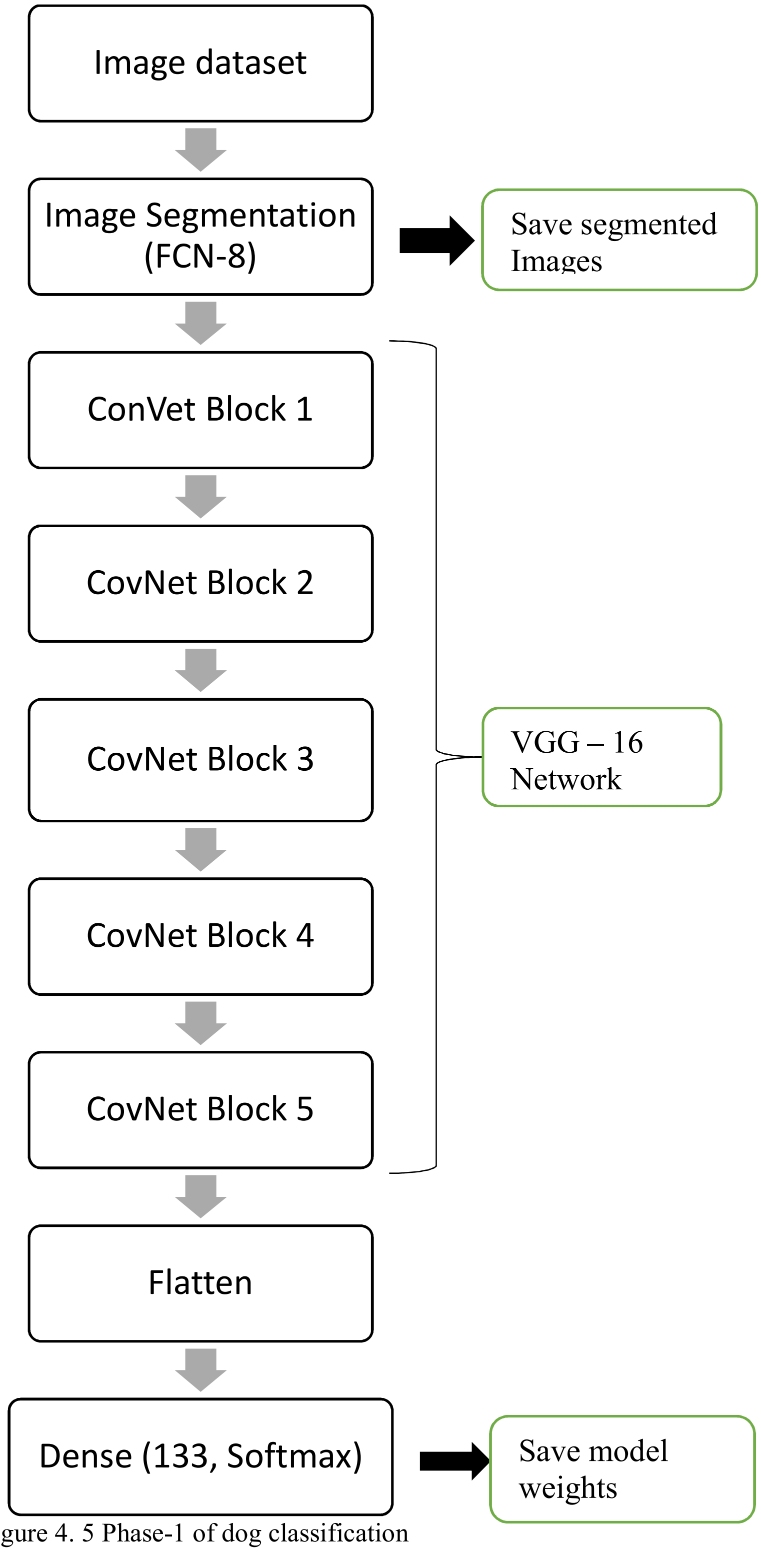


Figure 4. 5 Phase-1 of dog classification

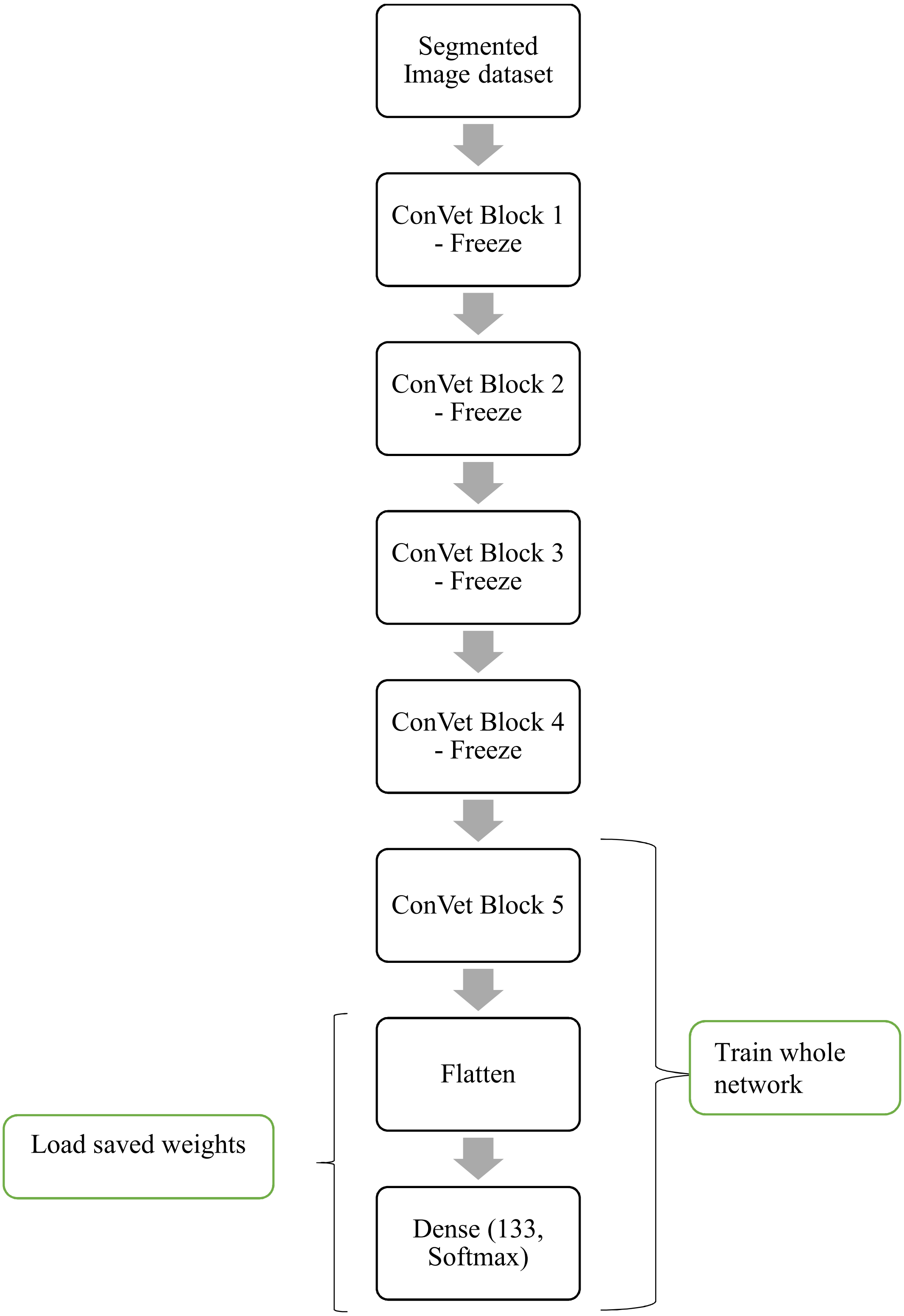


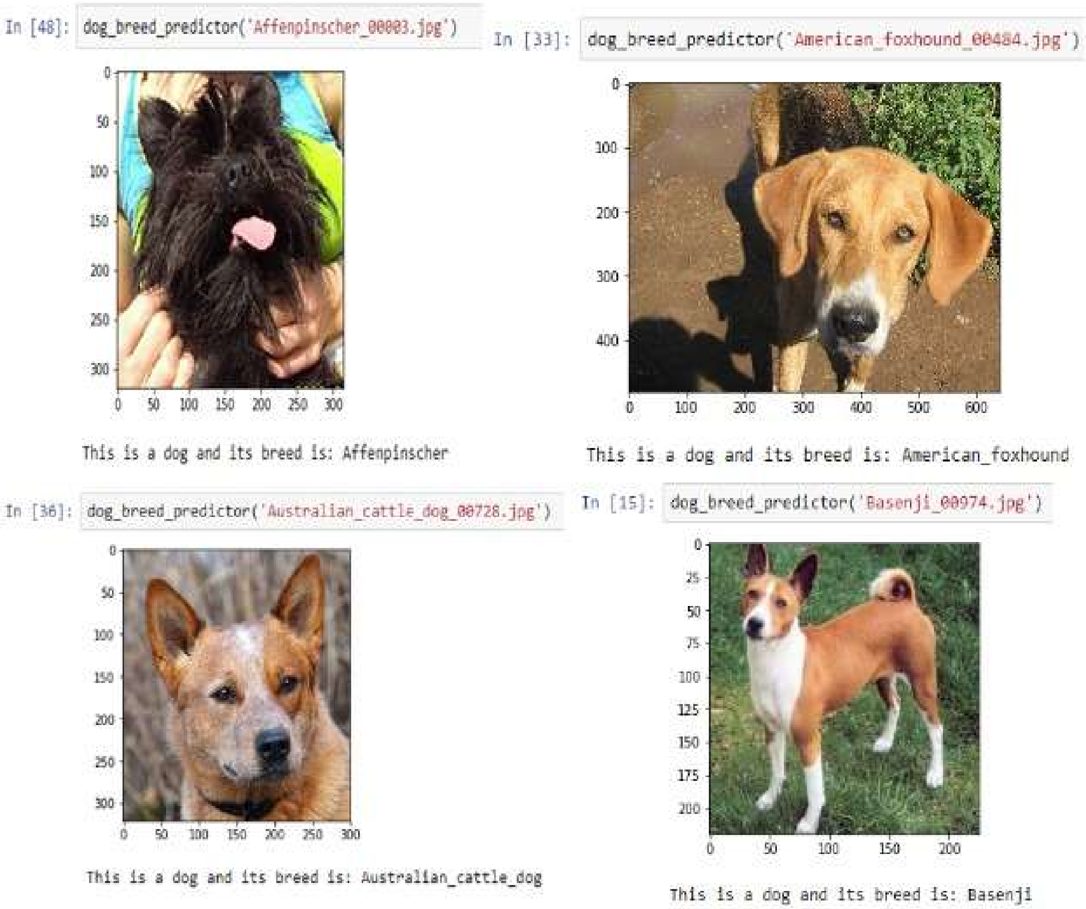
Figure 4. 6 Phase-2 of Dog Classification

S.No.	Method	Image Segmentation (Y/N)	Accuracy (%)
1.	VGG -16 model with a two-layer dense network	N	74.61
2.	VGG -16 model with a two-layer dense network	Y	86.34
3.	ResNet50 model with a two-layer dense network	N	82.46
4.	ResNet50 model with a two-layer dense network	Y	88.95
5.	Proposed Two – phase method	Y	97.12

Table 4. 3 Dog Classification

S. NO.	Method	Batch Normalization (Y/N)	Accuracy (%)
1.	VGG -16 model with a two-layer dense network	N	77.27
2.	ResNet50 model with a two-layer dense network	N	84.26
3.	Proposed Two-phase method	N	93.87

Table 4. 4 The difference of using Batch normalization in Dog classification



```
In [48]: dog_breed_predictor('Affenpinscher_00003.jpg')
This is a dog and its breed is: Affenpinscher

In [33]: dog_breed_predictor('American_foxhound_00484.jpg')
This is a dog and its breed is: American_foxhound

In [36]: dog_breed_predictor('Australian_cattle_dog_00728.jpg')
This is a dog and its breed is: Australian_cattle_dog

In [15]: dog_breed_predictor('Basenji_00974.jpg')
This is a dog and its breed is: Basenji
```

Figure 4. 7 Examples of correctly classified images

CHAPTER 5 CONCLUSION AND FUTURE WORK

Classification between similar images is a difficult task for an algorithm. However, through our proposed deep learning technique presented in this thesis. We have achieved the classification accuracy of 95% for 20 various types of bird species while got an accuracy of 97% for classifying between 133 different types of dog breeds.

To our knowledge, this work demonstrates the best bird and dog classification yet. Apart from other methods which use hand-crafted features, the proposed deep-learning approach learns the discriminative features through the convolutional neural network. Convolutional network for image classification extracts features from the images which allow the parameters to improve classification performance [26]. In this report, we have discussed the methods involved in fine-tuning a pre-trained neural network to efficiently classify the model to categorize the images with higher accuracy.

Also, describing an image segmentation technique to remove the background from the classifying subject. The experimental results performed on bird dataset and dog dataset work significantly better than other models to classify among different species and breeds which are similar in shape, luminance and having a complex background. The proposed methodology also states a novel method to train the model proficiently, making it robust against erroneous images consisting of multiple subjects.

Furthermore, we have used very little data to train the model to give high accuracy as shown on experimental results. The proposed two-phase algorithm trained twice with the same dataset through data augmentation technique which transforms the image to prevent overfitting. Augmenting the data improves the CNN classification by adding rotation on the images such that CNN attains robust learning.

The proposed algorithm can be extended in classifying between medical images due to the scarcity of sufficient images in training a neural network efficiently. In addition, the applicability of our algorithm on several kinds of classification problems which are similar to the bird and dog classification.

References

[1]. Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao and Li Fei-Fei. Novel dataset for Fine-Grained Image Categorization. First Workshop on Fine-Grained Visual Categorization (FGVC), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

[2]. Welinder P., Branson S., Mita T., Wah C., Schroff F., Belongie S., Perona, P. “Caltech-UCSD Birds 200”. California Institute of Technology. CNS-TR-2010-001. 2010.

[3]. M. Zhongli, L. Qianqian, L. Huixin, M. Zhongli and L. Zuoyong, "Image representation based PCA feature for image classification," 2017 IEEE International Conference on Mechatronics and Automation (ICMA), Takamatsu, 2017, pp. 1121-1125.

[4]. T. Tateyama, Z. Nakao and Y. Chen, "Classification of Brain Matters in MRI by Kernel Independent Component Analysis," 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Harbin, 2008, pp. 713-716.

[5]. S. K. Bidhendi, A. S. Shirazi, N. Fotoohi and M. M. Ebadzadeh, "Material Classification of Hyperspectral Images Using Unsupervised Fuzzy Clustering Methods," 2007 Third International IEEE Conference on Signal-Image Technologies and Internet-Based System, Shanghai, 2007, pp. 619-623.

[6]. A. T. Vo, H. S. Tran and T. H. Le, "Advertisement image classification using convolutional neural network," 2017 9th International Conference on Knowledge and Systems Engineering (KSE), Hue, 2017, pp. 197-202

[7]. Z. Zhou and Y. Zhang, "Integration of association-rule and decision tree for high resolution image classification," 2013 21st International Conference on Geoinformatics, Kaifeng, 2013, pp. 1-4.

[8]. H. Hiary, H. Saadeh, M. Saadeh and M. Yaqub, "Flower classification using deep convolutional neural networks," in IET Computer Vision, vol. 12, no. 6, pp. 855-862, 9 2018.

[9]. L. Zhang, Le Lu, I. Nogues, R. M. Summers, S. Liu and J. Yao, "DeepPap: Deep Convolutional Networks for Cervical Cell Classification," in IEEE Journal of Biomedical and Health Informatics, vol. 21, no. 6, pp. 1633-1643, Nov. 2017.

[10]. Very Deep Convolutional Networks for Large-Scale Image Recognition, K. Simonyan, A. Zisserman, available at: arXiv:1409.1556.

[11]. Deep Residual Learning for Image Recognition, Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, available at: arXiv:1512.03385.

[12]. Identity Mappings in Deep Residual Networks, Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, available at: arXiv:1603.05027.

[13]. A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," 2018 International Interdisciplinary PhD Workshop (IIPhDW), Swinoujście, 2018, pp. 117-122.

[14]. Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer, LNCS, Vol.9351: 234--241, 2015, available at arXiv:1505.04597.

[15]. J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 3431-3440.

[16]. Mask R-CNN, Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick, available at: arXiv:1703.06870.

[17]. Ioffe, Sergey and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.” ICML (2015).

[18]. J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 779-788.

[19]. F. Milletari, N. Navab and S. Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, 2016, pp. 565-571.

[20]. M. Chanvichitkul, P. Kumhom and K. Chamnongthai, "Face recognition based dog breed classification using coarse-to-fine concept and PCA," 2007 Asia-Pacific Conference on Communications, Bangkok, 2007, pp. 25-29.

[21]. X. Wang, V. Ly, S. Sorensen and C. Kambhamettu, "Dog breed classification via landmarks," 2014 IEEE International Conference on Image Processing (ICIP), Paris, 2014, pp. 5237-5241.

[22]. Guotian Xie, Kuiyuan Yang, Yalong Bai, Min Shang, Yong Rui and Jianhuang Lai, "Improve dog recognition by mining more information from both click-through logs and pre-trained models," 2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Seattle, WA, 2016, pp. 1-4.

[23]. D. R. Lucio, Y. Maldonado and G. da Costa, "Bird species classification using spectrograms," 2015 Latin American Computing Conference (CLEI), Arequipa, 2015, pp. 1-11.

[24]. B. Qiao, Z. Zhou, H. Yang and J. Cao, "Bird species recognition based on SVM classifier and decision tree," 2017 First International Conference on Electronics Instrumentation & Information Systems (EIIS), Harbin, 2017, pp. 1-4.

[25]. R. Narasimhan, X. Z. Fern and R. Raich, "Simultaneous segmentation and classification of bird song using CNN," 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, 2017, pp. 146-150.

[26]. M. Nilsback and A. Zisserman, "Automated Flower Classification over a Large Number of Classes," 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing, Bhubaneswar, 2008, pp. 722-729.

LIST OF PUBLICATIONS OF THE CANDIDATE'S WORK

[1]. Published a Conference paper titled "A Two-Phase Image Classification Approach using Very Less Data" on Springer Lecture Notes in Computational Vision and Biomechanics (Scopus Indexed).

[2]. Published a Journal paper titled "Copy-move forgery detection techniques: A literature survey" in International Journal of Engineering Technology Science and Research, ISSN- 2394-3386.\

[3]. Presented a paper titled "Copy-move forgery detection using Stationary Wavelet transform and SIFT" in ACSIT-2018 organized by Krishi Sanskriti, p-ISSN-2393-9907, e-SSN-2393-9915, pp-7-12.