ANIMAL RECOGNITION AND IDENTIFICATION USING MACHINE LEARNING

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Engineering degree in

Electronics and Telecommunication Engineering

by

MUTHUKUMARI M (Reg. No. 37250016)

RAMALAKSHMI M (Reg. No. 37250019)

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION 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 - 2021

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION

ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of Muthukumari M (37250016) and Ramalakshmi M (37250019)who carried out the project entitled " ANIMAL RECOGNITION AND IDENTIFICATION USING MACHINE LEARNING " under my supervision from November 2020 to April 2021.

Internal Guide

Mrs. T. GOMATHI M.Tech, (Ph.D.)

Head of the Department

Dr. V. VIJAYA BASKAR M.E., Ph.D.

Submitted for Viva voce Examination heldon 19/04/2021

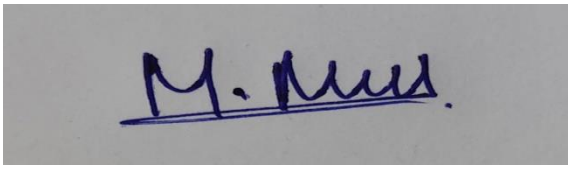
InternalExaminer **External Examiner**

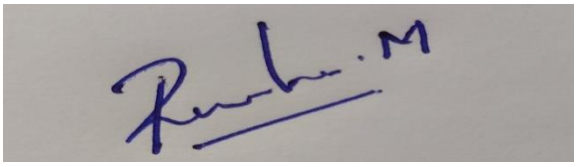
DECLARATION

I, **Muthukumari M(Reg. no. 37250016)**and **Ramalakshmi M(Reg. no. 3250019)**hereby declarethattheProjectReportentitled**“ ANIMAL RECOGNITION AND IDENTIFICATION USING MACHINE LEARNING ”**done by us under the guidance of **Mrs. T. Gomathi**is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and TelecommunicationEngineering.

DATE: 19/04/2021

PLACE: CHENNAI **SIGNATURE OF THE CANDITATE**

1. 

2. 

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for 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 and **Dr. V. VIJAYA BASKAR M.E., Ph.D.**, Head of the Department, Dept. of Electronics and Telecommunication Engineering for providing me 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 my Project Guide **Mrs. T. GOMATHI M.E., (Ph.D.)** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my projectwork.

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

I wish to express my thanks to the Project Panel members for their valuable feedbacks during the Project Reviews which were useful in many ways for the completion of the project.

ABSTRACT

Identification of wild animal in their environment is crucial. The proposed work develops an algorithm to detect the animals in the Wildlife. Since there are different types of animals it is a difficult task to identify them manually. This algorithm classifies the animals based on the images which are captured on the camera traps. This helps us to prevent Animal-vehicle accident, trace animals, prevent theft and we can reduce human animal conflict. This can be achieved by applying Capsule Network. The Wildlife conservation and human-animal conflict requires cost effective methods to monitor the animals. Still a video camera surveillance will generate enormous amount of data, which needs again lot of manpower and expensive to screen. In this project we have given a efficient algorithm known as Capsule Network for automatically identifying the animals from captured images.

INDEX

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
SIFT	Shift Invariant Feature Transform
SVM	Support Vector Machine
ReLu	Rectified Linear Unit
NiN	Network in Network
VGG	Visual Geometry Group
LASSO Operator	Least Absolute Shrinkage and Selection
GLCM	Gray Level Co-occurrence Matrix
LMD	Latin Music Database
LBP	local binary pattern
RWCP-SSD Scene	Real World Computing Partnership Sound Database
LPQ	Local Phase Quantization
CCAC	Canadian Council on Animal Care
MNIST	Modified National Institute of Standards and Technology
OpenCV	Computer Vision Framework Used for Pre-processing
SS	Snapshot Serengeti

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Surveillance of wild animals in the national parks and wildlife sanctuaries in a non-invasive approach in the ecological monitoring is a difficult task. Recent achievements in computer vision and Classification techniques including a deep learning allowed researchers to obtain the promising results. However, the recognition of animal species in the wildlife using camera traps remains as a unsolved problem due to many challenges, which are caused by the shooting conditions (like varying illumination, different weather, seasons and cluttered background) and animal or bird behavior (like unpredicted movement, multiple shapes and poses, occlusions by natural objects) [1].

Camera traps are on the predefined locations near the animal trails, watering places, salt licks, etc. Nowadays, the cameras which are kept for capturing cannot be connected into locally distributed networks with data transmission and communication in remote territories of the national parks and wildlife sanctuaries. The fixed position of the camera trap is determined under multi-years observations, and each camera trap is an automatic device with the motion and flash sensors, which stores and transmits information about the movement of the animal. As a result, each camera trap accumulates great amount of captured images or short movies, which are captured if any motion in a scene had been detected. The moving object can be animals, birds, or humans, but our interest is to detect and recognize the animals and birds excluding a human from the following analysis. Also, several moving objects may be detected during the particular photo session. If camera trap stores captured images, then several images for 3–5 s will be written in a hard drive as one event that is been detected. Each image is been automatically marked with the current date, time, and temperature values. As a result the set of images in the database can be sorted.

The analysis of dataset captured by camera traps in Ergaki national park, Krasnoyarsky Kray, Russia, 2012-2018, after excluding the non-recognized images (blurred images, images with low contrast, or non-understandable pose of animal) indicated that conditionally first sub-set contains good representation of the animal muzzles, second sub-set includes good representation of the animal shapes, third sub-set holds a part of theshapes, and forth sub-set of images involves the whole objects [1]. In this project, we propose the procedure, which categorizes the animal images, and architecture of the Images with a human are excluded during the categorization procedure.

1.1RELATED WORK

A conventional set of color-texture features are usually taken which are not variant to light and contrast variations, are proposed in the paper referenced. If animal types are previously known, some special features for the animal detection, recognition, and rectification, are used . Thus, for the individual identification of the marbled salamanders, skinks, and geckos, a feature of strong bilateral symmetry was additionally applied in [1]. In this peper, different types of matching like patch-based (multi-scale principal component analysis and scale-cascaded alignment), local feature-based (histogram, Shift Invariant Feature Transform (SIFT), and affine invariant variations), and context-based (hybrid shape-contexts with cascaded correspondence) matching, were used. In Conventional instruments like decision trees, fuzzy logic, genetic algorithm for the production of rule set, generalised linear models random forest and maximum entropy mehtod of machine learning approaches are used in species distribution modelling. Images which are manually cropped or selected, which contains whole animal shape will provide a high accuracy results. The results of combination of SIFT, cell structured local binary patterns, weighted sparse coding for dictionary learning, and linear Support Vector Machine (SVM) were reported. The Authors of the above mentioned work tested their method on a dataset consisting of 7,000 camera trap images of 18 species from two different field cites and achieved an average classification accuracy of 82% [2].

At present, deep learning algorithm is employed for the recognition in the wildlife as a state-of-the art technique . The most popular Convolutional Networks (ConvNets) for the recognition and classification tasks are the following [1].

AlexNet was a major milestone for the development of the deep ConvNets. AlexNet won the 2012 ImageNet challenge by a large margin. By creating nonlinear transformations AlexNet was the first CNN which has used Rectified Linear Network(ReLu). Network in Network (NiN) proposed in [8] was one of the first architectures, in which convolutions were implemented, in order to provide more combinational power to the features of the convolutional layers. VGG16 was composed of sixteen convolutional layers, multiple max-pool layers, and three final fully-connected layers.

To offer a high accuracy the GoogleNet architecture was designed to be computationally efficient by using 12 times fewer parameters than AlexNet respectively.

Residual learning model of ResNet was based on the connection between the output of one or multiple convolutional layers and their original input. Dense Convolutional Network (DenseNet) connected each layer to every other layer in a feed-forward fashion that led to better classification performance.

The Basic conceptual Convolutional Neural Network's various modifications are existing and continuing to appear.

All the above mentioned works consists of CNN architectures that were useful for identification, recognition and detection. The Overall species identification accuracy was 33.507% for the bag of visual words and the overall species recogniton accuracy of deep convolutional neural network was 38.215%.

The Deep Convolutional Neural Network contains three convolutional layers and three max pooling layers. The convolutional layer had a convolutional kernel with a size of 9*9, while the pooling layer had a kernel with a size of 2*2. The input layer size was 128*128, and the 3rd pooling layer in a view of 32 9*9 matrices

was transformed into 2,592 dimensional vector. The soft-max layer had almost 20 neurons and maximum output value among those 20 neurons determined input image label. Chen et al. employed a relatively small dataset of around 20 classes and 20,000 images. The upcoming investigations in this work were connected with deep CNN. Villa et al. studied AlexNet, VGGNet, GoogLeNet (Inception), and ResNets architectures for identification of the animal species using the camera-trap images. At that time, ConvNets were used as the blackbox feature extractors. The whole dataset Serengeti National Park, Tanzania was divided into the raw unbalanced images, raw balanced images, images with animals in foreground, and animals manually segmented from foreground. According to the ImageNet recognition challenge, the performance metrics Top-1 (the correct class is the most probable class) and Top-5 (the correct class is within the five most probable classes) provided the following results. This method reached 35.4% Top-1 and 60.4% Top-5 accuracy in the worst-case scenario with the unbalanced and empty images. At the same time, the accuracy reached 88.9% Top-1 and 98.1% Top-5 in the best scenario (the balanced dataset, images containing the foreground animals only, and manually segmented).

Nguyen et al. studied the Wildlife Spotter labeled dataset, Australia.

In that work they proposed the Wildlife detector with a CNN-based model which is been designed to train a binary classifier (with two classes, Animal and Non-animal) and Wildlife identifier as another CNN-based model created to train a multi-class classifier (species identification).

So called Lite AlexNet with less hidden layers and feature maps at each layer was applied as the Wildlife detector, while two ConvNets, VGG-16 and ResNet-50, were tested as the Wildlife identifier[1]. The suggested approach in that paper was allowed to achieve 96% for detection of animal and 90% for identification of three most common animals such as bird, rat, and bandicoot.

Sometimes, a combination is made of well-known methods created for classification of animals and segmentation using the camera-trap images. Thus, Giraldo-Zuluagaet employed several techniques for animal recognition in the Colombian forest, viz. the -layer robust principal component analysis for segmentation, CNN for extracting features,

Least Absolute Shrinkage and Selection Operator (LASSO) is explored for selecting the features, and artificial neural networks or SVM for classifying the mammal genera species. GoogLeNet, ResNet50, ResNet101, ResNet152, and MixtureNet were utilized as a CNN [3]. These authors achieved high accuracy values (for example, 92.65% classifying 8 mammal genera) with contribution not only CNN but also LASSO and SVM application.

Generally the most researchers use the datasets obtained from the Serengeti National Park, Australia, Tanzania, Northern America, or South-central Victoria without season changes as it's happened in Northern countries. We are the first, who study the animal recognition through all natural seasons in Russia, developing the combined background model [4].

CHAPTER 2

LITERATURE SURVEY

Currently, the animal detection and recognition are still a difficult task and there is no unique method to provides a robust and efficient solution for all situations. The animal detection algorithms implement by using animal detection as a binary pattern classification task [1]. That means, that given an input image, it is divided into small blocks and each block is transformed into a feature. Features from the animal that belongs to a certain class is used to train a certain classifier. Then, when given a new input image, the classifier will be able to check whether the sample is the animal or not. The animal recognition system can be divided into the following basic applications:

- Identification - compares the given animal image to all the other animals in the database and gives a ranked list of matches (one-to-N matching).
- Verification (authentication) - compares the given animal image and involves confirming or denying the identity of found animal (one-to-one matching).

While identification and verification often share the same classification algorithms, both modes target distinct applications [1]. In order to better understanding of the animal detection and recognition task and its difficulties, the following factors must be taken into account, because they can cause serious performance degradation in animal detection and recognition systems:

2.1 BASELINE METHODS

Since the main goal of our project is to evaluate the performance of CapsNet on different types of images (faces, traffic signs, everyday objects) we compared its performance with other suitable methods highlighted in the literature as reliable and robust for every type of datasets. First, we compare CapsNetwith the Fisherface algorithm [3] for face datasets (Yale face database B and MIT). This algorithm has proved to be fast, reliable [4,5], and one the most successful

methods for face recognition [6,7] achieving an average of 96.4% recognition rate on the Yale B Extended Database [8] and 93.29% on the MIT CBCL dataset.

For the Belgium Traffic signs dataset, we chose a CNN architecture known as LeNet-5 [9] and designed initially for handwritten digit recognition. The advantage of having a CNN is that it requires a much lower number of trainable parameters as compared to a Multi-Layer Feed Forward Neural Network, which supports sharing of weights and partially connected layers. Along with a reduced number of trainable parameters, a CNNs design also makes the model invariant to translation, a distinctive feature of state-of-the-art methods for image classification. LeNet-5 is relatively small but has proved to be powerful enough to solve several classification problems including traffic sign recognition [2].

A recent implementation of a LeNet classifier achieved 95% accuracy on the German traffic dataset [9]. Although we did not find any studies implementing LeNet on the Belgium TS dataset, a similar implementation of a simple Fully Connected network with two layers achieved 70% of accuracy [8]. Everyday objects classification is still a challenging computer vision problem. One of the state-of-the-art methods for this task is called Deep Residual Learning for Image Recognition (ResNet) [3]. It was presented in configurations with 18, 34, 50, 101, 152 layers. The best top-5 error achieved on ImageNet test dataset (1000 classes of different objects naturally appearing in the world) [4] is 3.57% with an ensemble of six ResNet.

CHAPTER 3
SYSTEM DESIGN

3.1 EXISTING SYSTEM

Efficient and reliable monitoring of the Animals in their natural habitats is a difficult task. So automatic cameras such as “Camera Traps” are been used nowadays which became popular tool for wildlife monitoring since it should be monitored unobtrusively, continuously and it will be in large value. However processing of those collected data is cost effective, requires manpower, time consuming and monotonous. This is a major drawback for scientists to monitor the animals in an open environment. The Enormous amount of data from the camera traps is highlighting the need for image processing automation. To resolve this problem Machine learning algorithms are used. In Machine Learning there are immediate techniques to identify the animals such as Support Vector Machine (SVM), Convolutional Neural Network (CNN), ImageNet etc. These approaches has addressed the problem of Wildlife monitoring Automation.

In the existing system Convolutional Neural Network is used for the Classification of Image. In the area of Image classification CNNs showed a greater performance and was been widely used in machine learning. Especially in the areas of classification of image, natural language processing and speech recognition. It was initially proposed by LeCun et al. The Models have made good performance in the human in the task of image recognition. Due to the improvements in the neural networks such as deep CNNs, the success particularly in the implementing the parallel computing power, and in the Tensorflow in large scale which is the heterogeneous distributed systems in the deep models. CNNs are the neural network model particularly designed to take the spatial structure of the input images, which has a three dimensional value : depth, height, width which are the number of color channel

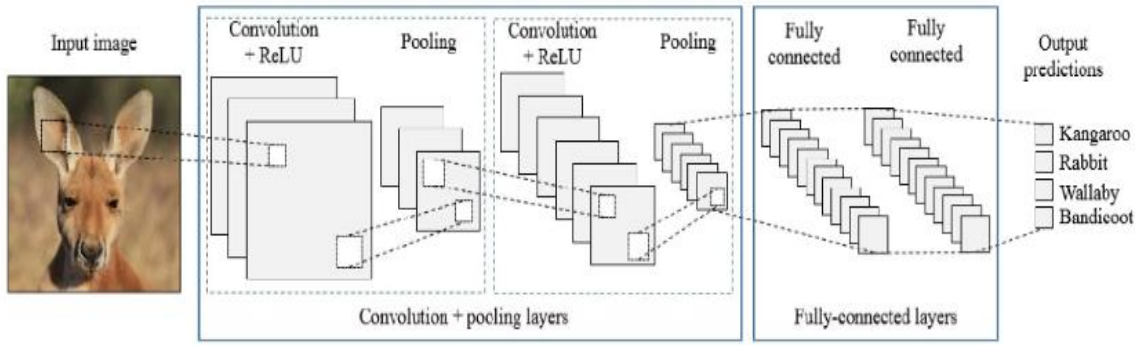


Figure: 3.1 Process of CNN

As shown in the above figure, the Convolutional Neural Network is the sequence of layers which can be categorised into groups each having a convolutional layer with a non-linear activation funciton. Generally there will exists the Pooling layer, the Rectified Linear Unit (ReLu) and mostly the Max pooling; This will be resulting in the fully connected layers where the last one is the output layer with the predictions of the image. In the standard neural networks, the neurons in each layer are completely independent and each neuron is fully connected to all neurons in the previous layer. The total number of parameters can reach millions, leading to serious over fitting problem and it will be impractical to be trained when applied to high dimensional data such as natural images. By contrast, each neuron is connected only to a small region of the preceding layer forming local connectivity in CNNs. The convolution layer computes the outputs of its neurons, the spatial extent of this connection is specified by a filter size which are connected to local regions in the previous layer. Moreover, dramatically reduces the number of parameters and so does computing complexity namely parameter sharing is the important property. CNNs have fewer connections and parameters, making them easy to train while their performance is slightly degraded; thus, it is having compared to regular neural networks with similar size of layers,.

Converting input image into layers of abstraction is done by three main characteristics. They are

- ✓ Parameter sharing

- ✓ Local connectivity
- ✓ Spatial strucutre.

The higher layers exhibit - more abstract features of object while the lower layers - present detail features of images such as edges, curves and corners.

Apart from using better techniques for preventing overfitting and more powerful models, the performance of machine learning which is been data driven approaches will depend striclty on the quality and size of the collected training datasets. Requiring much larger training sets to learn recognizing them, the real-life objects exhibit considerable variability.

3.1.1 DRAWBACKS

- Sub-sampling process loses precise spatial relationship information between higher level features such as nose and mouth which are required for identity recognition.
- And also, CNNs do not store relative spatial relationship between features. CNNs cannot extrapolate the understanding between the geometric relationships to radically view points.
- CNNs uses pooling layers to reduce parameters so that it can speed up computation.In particular during Max-pooling most of the valuable informations are lost.
- CNNs will not give haigh accuracy in test dataset unless trained with huge amount of dataset.
- CNNs basically try to achieve “viewpoint invariance”.

3.2 PROPOSED SYSTEM

- The purpose of animal detection systems is to prevent the accidents due to animal-vehicle collisions. This results to death, injury and also property damage for humans.

- Wildlife-vehicle and Wildlife-Human encounters often result in injuries and sometimes fatalities. Thereby, this work aims to diminish the negative impacts of these encounters in a way that makes the environment safer for both humans and animals.
- For this project, the main focus is going to be on wild animal data and prove that capsule network can do better and faster the task than human equivalent labor.
- Incidentally, we are going to take advantage of these data to make a comparative study on multiple deep learning models, specifically, VGG-net, RES-net, and a custom made Convolutional-Capsule Network
- Attempts to identify and count the animals wild animals with camera traps, explore some state of the art models and obtained some quite interesting results.
- A Capsule network is a neural network that does an inverse graphic to learn features from an image.
- In Computer vision, inverse graphics is a concept that allows the identification of objects in the image and helps to find the instantation parameter of the object.
- Capsule network effectively uses inverse graphic to produce a multi-dimensional vector that in capsulates various components of an object

3.2.1 BLOCK DIAGRAM

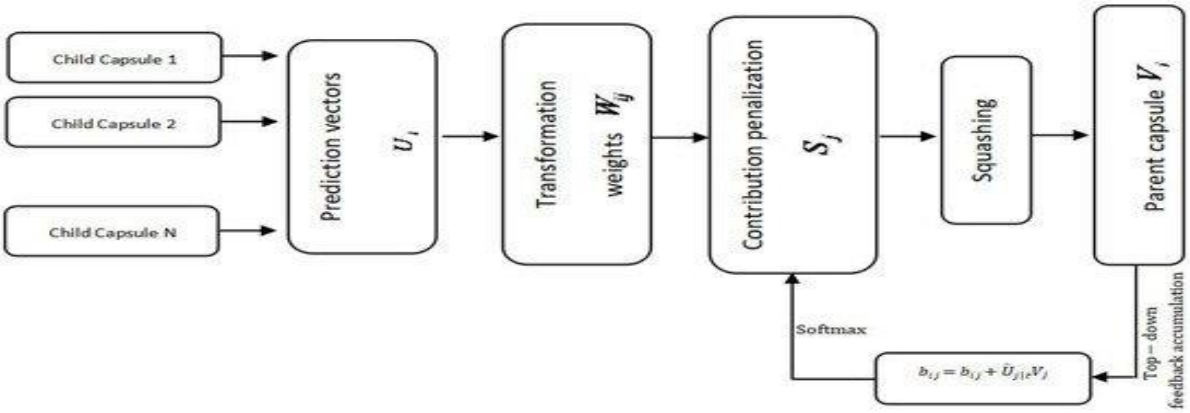


Figure 3.2 : Block diagram of Proposed methodology

3.2.2 PROJECT IMPLEMENTATION

- The main goal of project is to be able to automatically identify animal in the wild. An Advantage is been taken incidnetally in this project, by exploring the capsule networks in large and complex dataset respectively.
- That is why our comparative study between capsules and Convolutional neural network will be the second axis of our research. To exhaust and covert the two axes of the research, the first one to be explored is key pre-processing techniques and the state-of-the-art architectures to help us acheiveoujr main goal. This Attempt will provide a strong base to initiate as to which will be countouring factor in the same task that will ameliorate state of the art models.
- We should note that we are looking at full automation, so we will be more interested in top-1 accuracy all along the research. Moreover, explore capsule networks; by engineering custom models and comparing the performances.

- As our choices of network has been based on performances and diversity. We are selecting Capsule network as our 3 type of model to explore. In point of fact VGG was very deep nettwork with very small 3*3 filter. Inception is the network followed by it showing sparsity and parameter efficacy in the simultaneous convoluition on the top.Capsule network comes with a completely different way of learning from images
- The First one will be with a 3-layer network and a very low-resolution image. This will be to explore the efficiency of capsule on low resolution image and the effect of image alteration on capsule network performances.
- The second one will be a more customize Capsule network with 11 layers. Here we are going to use a more complex image and try to explore how our custom build network can perform on a more complex image.

3.2.3 ARCHITECTURE

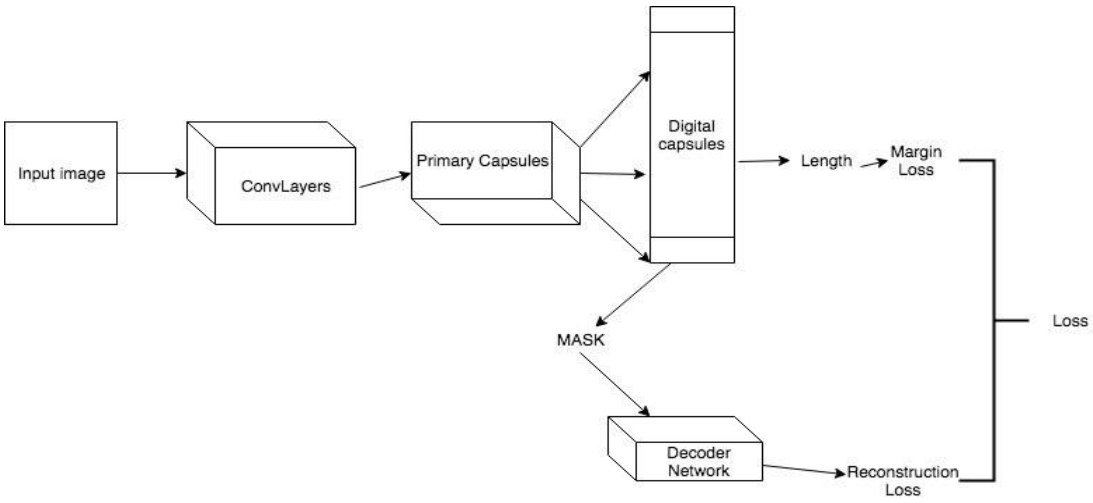


Figure 3.3 : Architecture of Capsule Network

- **Matrix multiplication -** Matrix multiplication is applied to the image that is given as an input to the network to convert it into vector values to understand the spatial part.

- **Scalar weighting -**Scalar weighting on the input computes which higher-level capsule should receive the current capsule output.
- **Dynamic Routing Algorithm -** It permits these different components to transfer amongst each other. The Lower level capsules gives the input to the higher level capsules. This is a repetitive process.
- **Squashing Function -** The last component that condenses the information is the squashing function. The Squashing function converts it into a vector by takes all the information. The vector will be less than or equal to 1 also maintaining the direction of the vector.

A Capsule network is a neural network that does an inverse graphic to learn features from an image. The Concept known as Inverse graphics in computer vision which allows to identify the objects in the image and the instantation parameter of the object. Multi-dimensional vector that incapsulates various components of an object is produced with the use of effective inverse graphics which is used by capsule network. The thickness, the dept, the length, shape, localized skew, localized parts, and many other variants of an object. Many other variants of the body including the thickness, the dept, the length, shape, localized skew, localized parts.

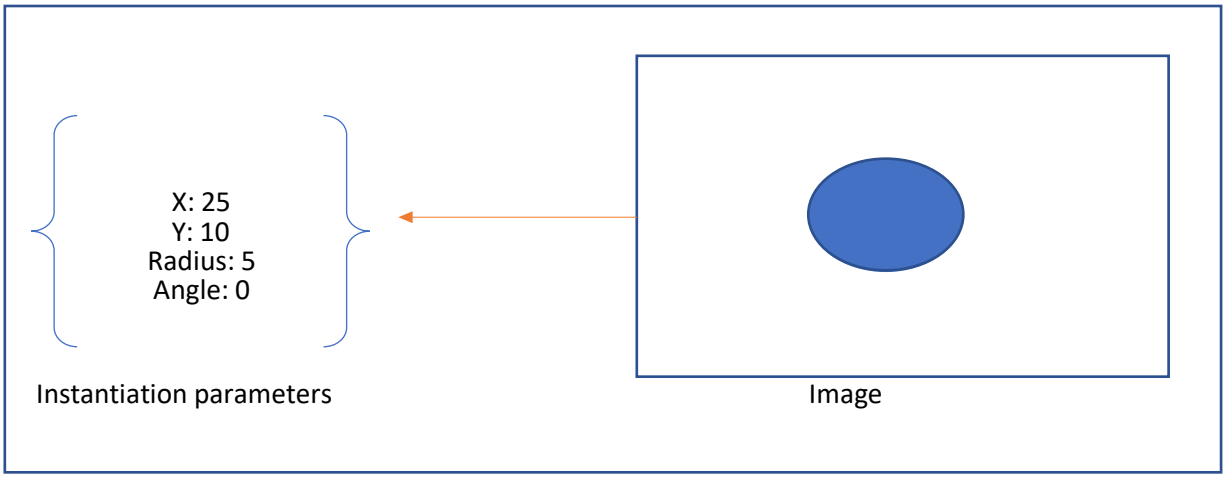


Figure 3.4 : Inverse graphic example.

Capsule is the key component of the capsule network. A capsule is a function that tries to identify any given instantiation parameter of an object. A capsule is composed of an activation vector. It typically has two dimensions:

- the probability of the object embodied by the capsule to be detected is represented by the length.
- the orientation, that characterizes the pose parameter of the object embodied by the capsule.

Anyhow, Activation vectors may have configuration of the capsules and many more dimensions depending on the goal.

In spite of this fact, it can be get back on one of the dimensions of the activation vector, the length. The length is representative of the probability of an object being present based on its instantiation parameters. It should always be less or ideally equal to 1 is meant. For this purpose, a new activation function was designed for the capsule. The so Called, Squash function. To ensure that the activation vectors are always less than 1 the squash function is designed.

In addition to that, the first layers of capsules, predicts the output of the next layer from every capsule. This means, during training, the network will try to learn the transformation matrices from one vector in the first layer to another one in the next layer. What does this actually mean?

Let suppose capsule network that is supposed to correctly classify different kinds of objects amongst which we have a bicycle and a vehicle. One that detects andidentifies the instantiation parameters of a wheel oriented amongst the capsules of the first layer. The output is, a bicycle and a vehicle respectively oriented at the same angle as the first layer circle capsule where capsule will predict that two of the next layer's capsules. Another example will be that of a house and a boat to be predicted. A capsule detecting the Triangle component of either the boat or the House amongst the first layer of capsules (primary capsules). If that capsules effectively detects a triangle, it will predict the output of the capsules in the next layer. A House oriented at an angle

similar to its primary capsule will be obtained as output from the house capsule which is similar to the boat capsule.

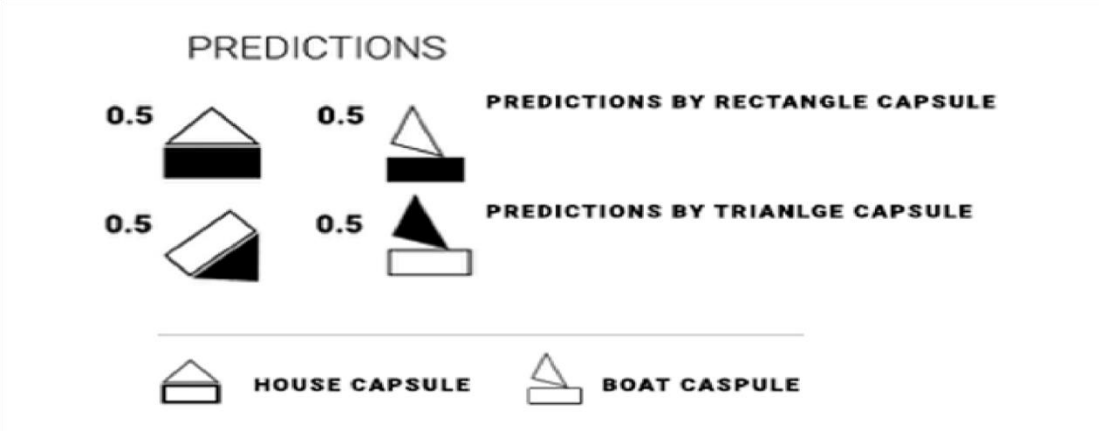


Figure 3.5 : Prediction

After each capsule in the primary capsules, layer has contributed to the construction of the next layer, the capsules of the last capsules layer have to decide what they actually identify. It is a democratic process between capsules. Meaning that the majority of the capsules have to agree on what output should be. This process is called routing by agreement. In the point of fact, all the capsules will depend on the inputs and predictions of the previous layer, in the last capsule layers, vote on what is the most suitable object detected in the given image. This vote is based on all the instantiating parameters the capsules were set to detect.
Indeed, if we have to predict between a house, a boat and a car, each primary capsule will vote as to what is supposed to be output based on what they identify. The object having the highest vote will be the one predicted. When all the capsules involved in the process all agree on the same object, we talk about Routing by agreement.

Strong agreement.

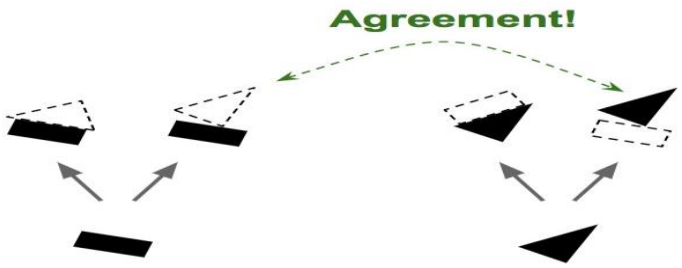
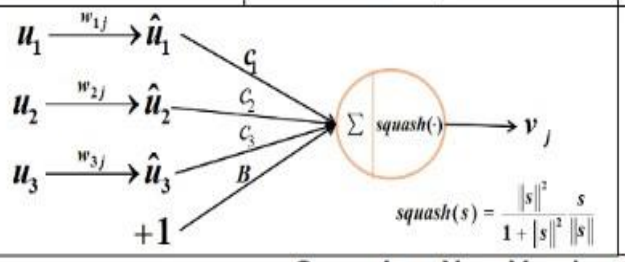
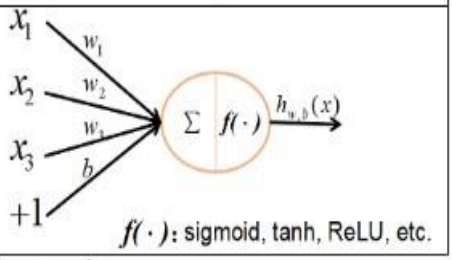


Figure 3.6 : Agreement

All these features and mechanism behind the Capsule networks allow them to preserve at least the location and pose of objects throughout the network. Hence, Capsule network is Equivariant. We will use these key concepts to build custom networks and perform several experiments with capsules-based networks.

The table below resumes a brief comparison between capsules network and

		capsule VS. traditional neuron	
Input from low-level neurons/capsules		$\text{vector}(u_i)$	$\text{scalar}(x_i)$
Operations	Linear/Affine Transformation	$\hat{u}_{j\|i} = W_{ij} u_i + B_j$ (Eq. 2)	$a_{j\|i} = w_{ij} x_i + b_j$
	Weighting	$s_j = \sum_i c_{ij} \hat{u}_{j\|i}$ (Eq. 2)	$z_j = \sum_{i=1}^{3} 1 \cdot a_{j\|i}$
	Summation		
	Non-linearity activation	$v_j = squash(s_j)$ (Eq. 1)	$h_{w,b}(x) = f(z_j)$
output		$\text{vector}(v_j)$	$\text{scalar}(h)$
			

Capsule = New Version Neuron!
vector in, vector out VS. scalar in, scalar out

the traditional networks

3.2.4 Intuition behind Capsule Networks

Figure 3.7 : Picture of Animal - Cat

Case 1 : Identification of Cat

For instance a picture of Cat is taken. But how it is identified that it is cat ? The General Approach is by seeing their features. The Individual features such as eyes, nose, ears etc. are broken down from the picture.

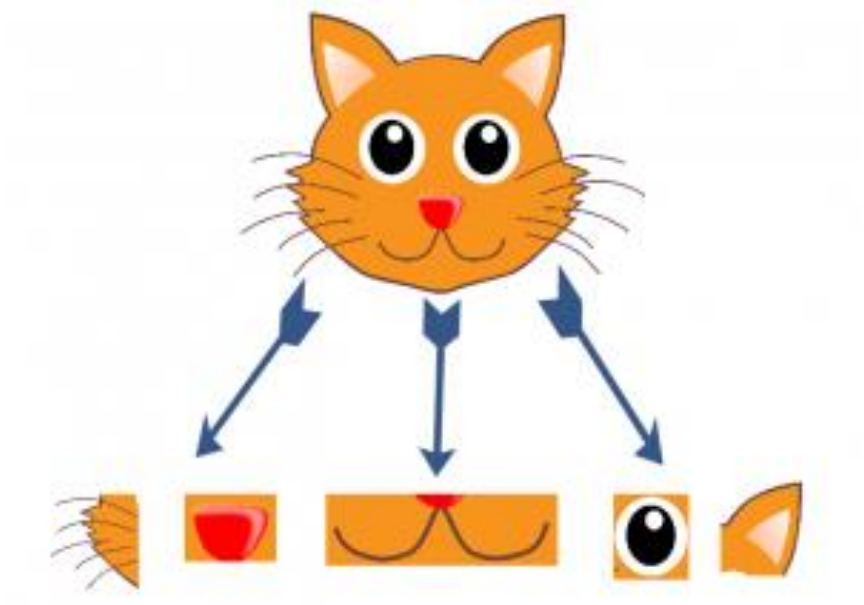


Figure 3.8 : Individual Features of Cats

Here the essential high level features is been decomposed to low level ones. This can be defined as,

P(face) = P(nose) & (2 x P(whiskers)) & P(mouth) & (2 x P(eyes)) & (2 x P(ears))

Where P(face) is defined as the presence of face of the cat in the picture respectively. It can also been done for low level features like edges and shapes in order to decrease the complexity of the procedure.

Case 2 – Rotated Image

The Second case is that for a rotated image,

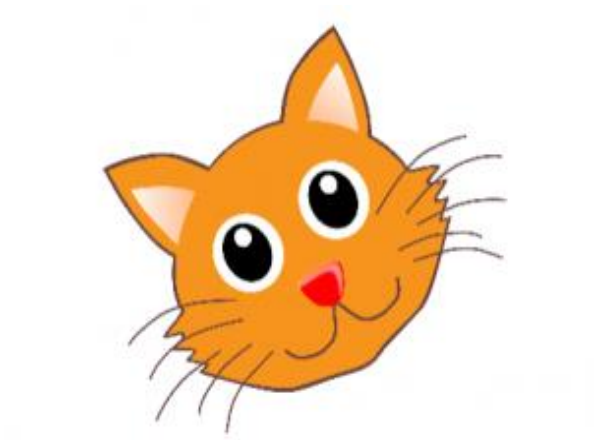


Figure 3.9 : Rotated Image of Cat - 90 Degree

Figure 3.10 :Rotated Image of Cat - 180 Degree

In this case the process cannot be taken through like extracting their features because if the image is rotated, the identification of cat is not possible. Since the Orientation of the low level features is also changing in rotation the feature extracting method cannot be undergone here since the predefined features will not match to the rotated ones.

So now, the low level features taken from images is depicted below.

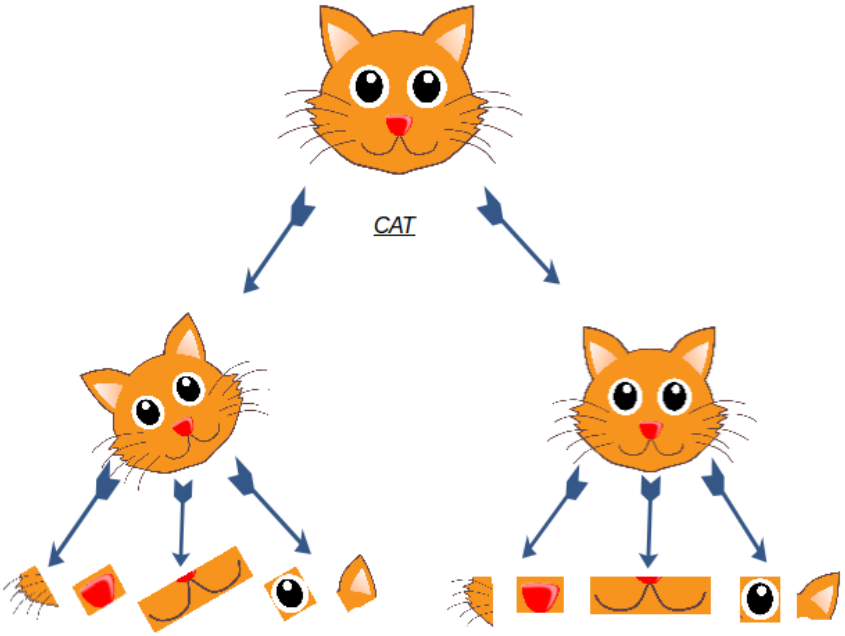


Figure 3.11 : Comparison Extracted features of rotated and proper image of cat

In this case, the better approach can be done is the Brute Force Search. Brute Force generally means finding for all possible combinations. The Low level features are rotated in all possible rotations and the feature matching is done.

Including additional properties of the low level features itself like rotational angle was one of the main suggestion by the researchers. Although it's rotated this is the way how the presence of feature can be checked. For instance, notice the below given picture.

In a more rigorous way, it can be defined as:

The Rotational value of the individual R() of a feature is been obtained. In a meaningful way for this approach the change in rotational value is represented. And this the rotational equivariance property. To capture more aspects of the low level features, such as skew, stroke, thickness, scale etc. this idea can be scaled up. With the help of this the image can be more clearly grasped. When they are designed this is how capsule networks are envisioned to work.

These are the important aspect of Capsule networks. Dynamic Routing is the another important feature of Capsule network. Another example is taken to explain that process.

Now, a picture of dog and cat is taken for the classification problem.

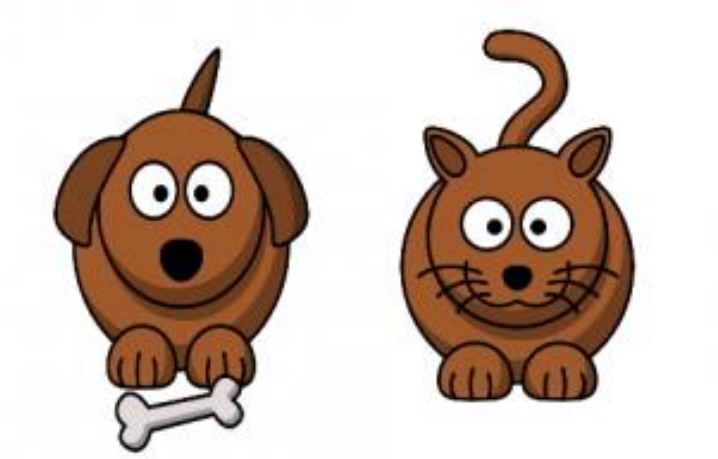


Figure 3.12 : Similar picture of Dog and Cat

They are very similar if seen entirely. But to figure out them as cat and dog there are some significant features in the images which help us identify them.

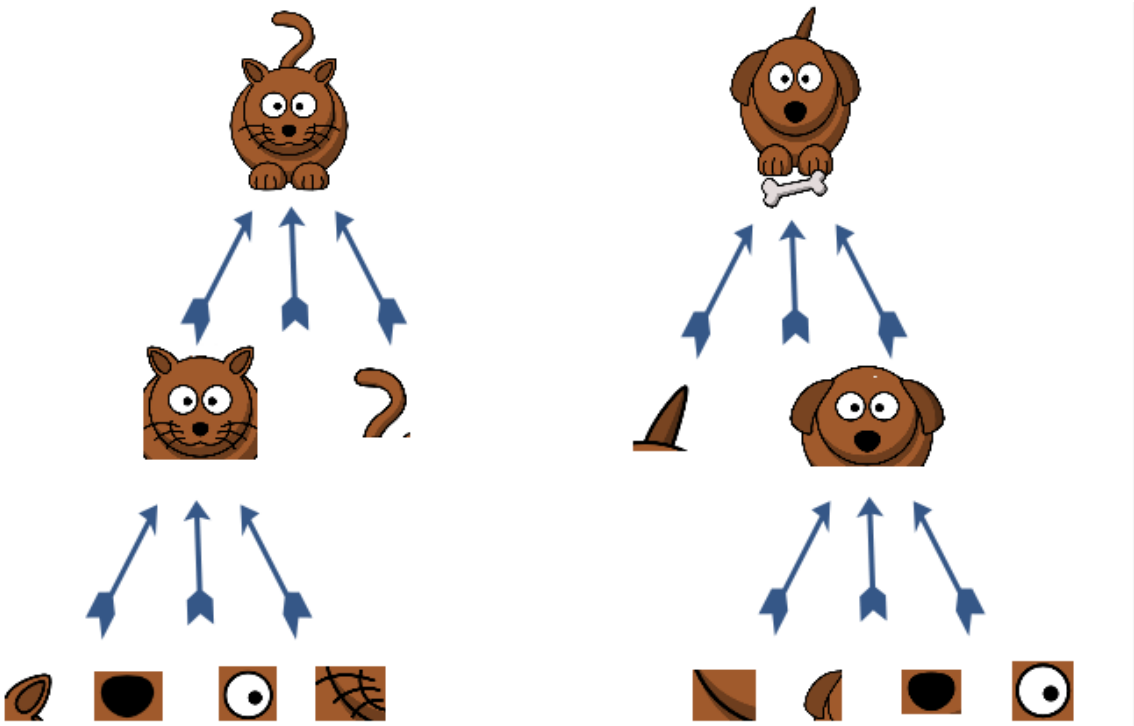


Figure 3.13 : Comparing Feature of similar picture of Dog and Cat

As done before in the previous sub-section the features in the images can be defined which will help us identify them.

The Complex features can be iteratively defined which come up with the solution as seen in the image below. Initially the very low features like eyes and ears are defined and then combine them to identify the face. Following that the facial and body features are combined to arrive at a solution as the given image is dog or cat.

If a new image is taken with the low level features extracted. The Class of this image is tried to figured out.

A Feature is been randomly picked and it is eye actually. There arises a question that can it only be enough finding the class?

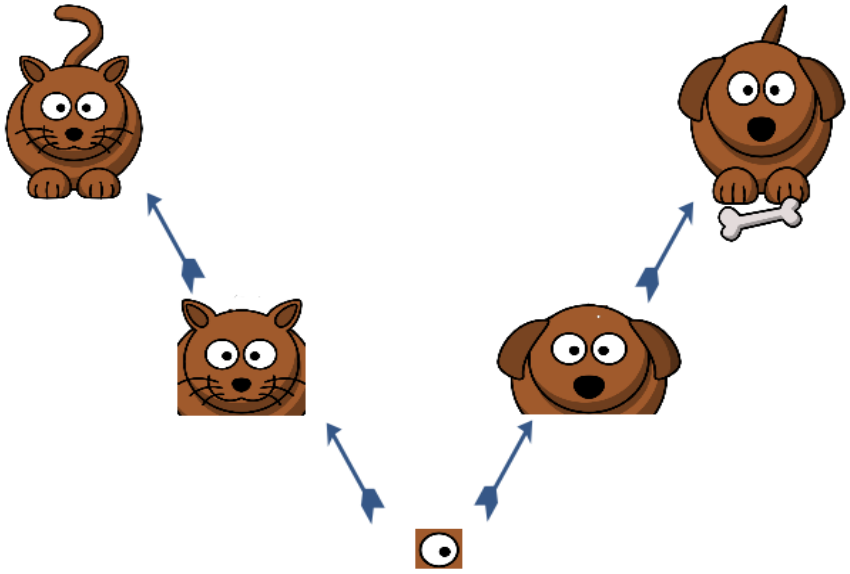


Figure 3.14 : Compering it with the one feature - Eye

Yes, it is noticed that eye alone is not a differentiating factor. And so the next step is to add more features to the process for analysis. The Next feature which is randomly picked is the nose. For this moment only the intermediate level features and low level features are seen.

.

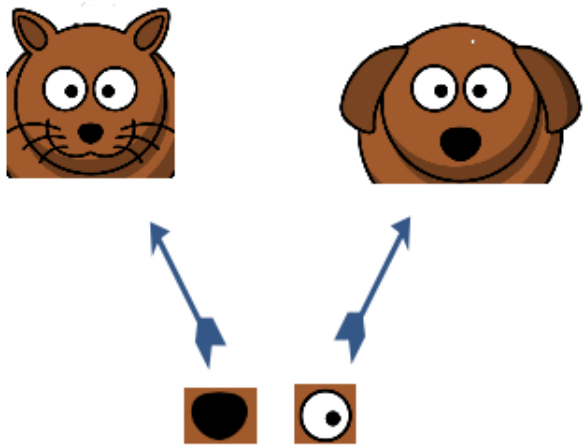


Figure 3.15 : Compering it with the two features - Nose and Eye

Still it is seen that it is not enough for classification. Now the next step is to consider all the features, combine them, guess and the output is taken by estimating the class. In this example we see that eyes, nose, ears and whiskers are combined. It is seen that it is more probable that both cat's face rather than a dog's face. So more weightage is given to the features when performing cat recognition in the given image. This step is iteratively done at each feature level so that correct information can be routed to these feature detectors that need the information of class.

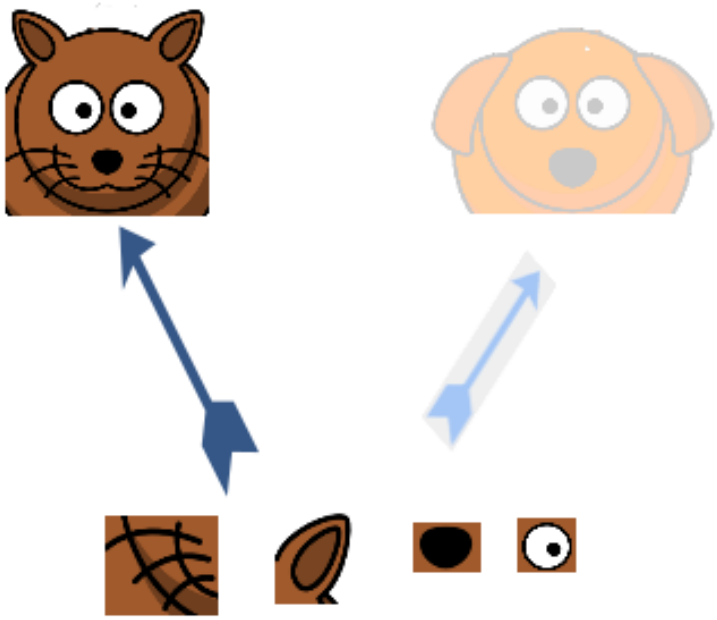


Figure 3.16 : Comparing many Features

Simply the each lower level, the most probable output at its immediate higher level is been tried to figured out. Now the question is, will the high level feature activate when it gets the information from the features. If it's indeed activate the lower level feature will give the information to the higher level. If it is somewhat irrelevant for that feature detection it will not pass on the information.

In the terms of Capsule, the higher level capsule will get the information from the lower level that agrees with the input. This is essence of dynamic routing algorithm.

These are the most important essential aspects of the capsule network, which makes it stand apart the the traditional deep learning architectures - dynamic routing and equivariance.

The Result of the process in capsule networks is that it is more robust to orientation and pose the information and it can be trained on less number of data points solving the same problem with better performance. A State of the art performance of capsule networks is developed by the researchers on the ultra-popular MNIST dataset with a couple of hundred times less data.This is Capsule Network's power.

Although these capsule networks come with their own difficulties like requirement of more training time and resources than the deep learning architectures in comparison. But this is just a time of matter before it is figured out how it can be tuned and come out of the production from current research phase.

3.2.5 ADVANTAGES

- “Equivariance” is tried to acheived by Capsule . It means output will change by changing input a little bit but length of vector will not change which will predict the presence of same object.
- It saves spatial relationship between features and also less amount of data is enough to be trained in case of Capsule Network.
- Capsule network do not use pooling layers which removes the problem of loosing useful spatial featured information.
- Initially capsules proceed with the matrix multiplication of the input vectors with weight matrices which actually tells us in detail about the spatial relationship information of some high-level features with low-level features.

- Dynamic routing is used for the selection of parent capsule. Parent capsules are decided by the capsules.

CHAPTER 4

EXPERIMENT AND RESULT

The main goal of our project is to be able to automatically identify animal in the wild. Incidentally, we take advantage of this project to create the opportunity of exploring capsule networks in complex dataset and large. That is why our comparative study between capsules and Convolutional neural network will be the second axis of our project.

To covert and exhaust the two axes of our project, we will first explore state of the art architecture and key pre-processing techniques to help us achieve our main goal. This will give off a strong basement to start as to what will be the contouring factor that will ameliorate state of the art models on the same task. We should note that we are looking at full automation, so we will be more interested in top-1 accuracy all along the research. Moreover, explore capsule networks; by engineering custom models and comparing the performances.

After gathering all the performances of our multiples experiments, we will first benchmark our work with several results obtain from people that performed a similar task, then compare our performances between each other. More precisely, we will first compare our result with recent research that attempted this kind of classification task using state of the art models. Then we will compare our capsule network results with that of research that have attempted it with large and complex dataset. Finally, we will compare our own results with each other and get the best out of it. This will allow us to give a prospective analysis as to what to improve and what are the shortcomings of using one methodology over the other.

As our choices of network has been based on performances and Diversity. We are selecting Capsule network as our 3 type of model to explore. we saw VGG as very deep network with very small 3 by 3 filter. Followed by the Inception which was network advocating sparsity and simultaneous convolution on top of the parameter efficacy. Capsule network comes with a completely different way of learning from images. Instead of dealing with pixels, it's deals with instantiation vectors as we discussed in Chapter II.

Capsule network theory provides some confidence to image classification as the models are built to be rotation invariant and performs inverse graphic to ensure that the proper vectoral features of an object in an image are extracted. However, the best result achieve with Capsule network was a 0.25% error on the MNIST dataset. Motivated by the promising future of this new techniques we decided to explore it on more complex dataset compare to MNIST.

We are going to perform two set of experiments on Capsule network.

- The First one will be with a very low-resolution image and a 3-layer network. This will be to explore the efficiency of capsule on low resolution image and the effect of image alteration on capsule network performances.
- The second one will be a more customize Capsule network with eleven layers. Here we are going to use a more complex image and try to explore how our custom build network can perform on a more complex image (300by300).
- For each of our Capsule network, we will perform image reconstruction to minimizing the changes of overfitting and to ensure that our network actually learns the general patterns in the images. For experimentation the details of these experiments will be given in the next chapter.

4.1DATASET

We have used a Complex Dataset which can identify twelve animals. Due to the exhaustive nature of the capsule network and limited resources, we had to create a significant subset of this dataset. We noticed a huge data imbalance in the distribution of classes where few classes accounted for most of the images. Hence, we decided to go with the 12 most occurring animals for this project: Elephant, Lion, Tiger, Bear, Zebra, Deer, Monkey, Fox, Cow, Pig, Rabbit and Squirrel . For these experiments, we split the data into training, testing and validation. 80% of the data went towards the training and validation set, and 20% went towards the testing set. Out of the 80% allocated to testing and validation, 20% of that went towards validation and the other 80% was purely training set.

4.2 PREPROCESSING

Pre-processing has been a keep part of the work we did here as it allowed us to have better models, and it increased the performances from one operation to another and it eased our workflow. We identified a few things that might be a problem for our experiment.

The first thing was to identify the class distribution from the metadata file in the Serengeti official website. Once that was done, we selected the 10 most occurring classes which happened to account for a good portion of the images. Then, we need to prepare a script to download these images and classify them by folders based on their label. This operation really eased the processed when I came to input the data into the models.

Secondly, the size of each image was big due to the resolution of each image. Keeping 1K(1,920x1,080) images would be computationally expensive for each and every single operation on every single image. As a solution, we decided to resize all the image to 300 x 300. 300 x 300 because most of the state-of-the-art model needed less than this to achieve a good result. For example, Inception which requires the biggest image size, use 299 x 299 images as an input.

The next issue faced was that some images were in the day, others were in the dark. We emitted a hypothesis that this could bias the model in case there is no balance between day and night or the model will tend to learn more about one versus the other. To solve this, we did a colour alteration on all the images, transforming them to grayscale and then applying another grayscale transformation. This will remove the presumably biased on all the images.

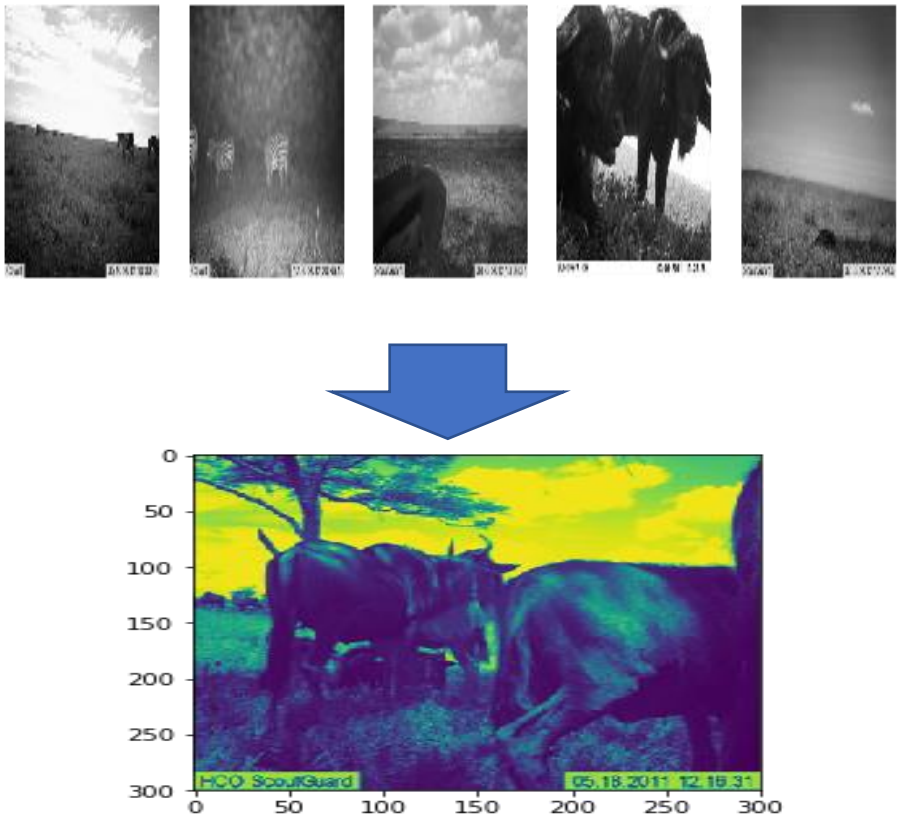


Figure 4.1 : Example of double Gray-scale Transformation

The first set of images represent the first transformation applied, where we convert all the images to Gray-scale once while keeping the 3 colour channels.

The second image represents the next issue faced was that some images were in the day, others were in the dark. We emitted a hypothesis that this could bias the model in case there is no balance between day and night or the model will tend to learn more about one versus the other. To solve this, we did a colour alteration on all the images, transforming them to grayscale and then applying another grayscale transformation. This will remove the presumably biased on all the images.

second grayscale transformation that eliminates any biased linked to the time of the day or the weather.

This transformation also enables us to get rid of our third problem, the weather. These transformations will allow the model not to be biased by the weather but to focus only on the shapes and hard colour variation (intensity) of the animals.

After these basics' transformation on the image, we had to reshape the images again before passing them to each model. These were necessary because each of the models needed different input sizes from 28 by 28 to 300 by 300. During this operation, we also performed batch normalization to all our images before inputting them to our model. There was a key process because of the large amount of data we had, the size of each image and the complexity of our models. Normalization allowed us to run our model faster while keeping the stability and reducing the chances of overfitting.

Uniform aspect Ratio was also a key component of our pre-processing as we had to make sure that every time, we scale down the images, the ratios between the height and the width is a 1:1 ratio.

4.3 ENVIRONMENT

The type of resources we used for this research was key factor in the achievements or no achievements we have had so far. In terms of hardware, we used two different devices.

- The first one was a Mac Pro, with a CPU of 3.5ghz 6-core intel Xeon E5, two graphic cards from AMD having 3 Gb of memory each; 32 Gb of RAM and a storage of 512Gb SSD.

  This was the initially hardware used at the early stage of our research. It helps us investigate the pros and cons of most of our model and perform the literature review. However, when we tried to start the actual experiments, this hardware was quite limited for the following reasons. The Storage on the device was not big enough to hold the dataset; there was also no GPU on the devices which did not allow us to use Cuda and the parallel processing abilities of the graphical processing unit. Finally, it

served as a terminal station to ssh into the Main server that we are presenting below.

- The second hardware was a server hosted by Kennesaw State University and managed by the College of Computing and Software Engineering. It had 1 Tb of storage allocated to me and **4 Tesla M40 GPUs.** This is the device that allow us to host our data and perform all our experiments to an extent allow by our computational limitations.
- The Software environment used for this research was comprised of the following tools: oOSX Yosemite : Operating System hosted on the Mac pro. oLinux server : Operating System of the CCSE server where the GPUs are hosted.
- FileZilla: File Transfer and SSH software used to graphically interface with the server's

  file.

- Jupyter Notebook: Integrated development environment used to build and visualize the models.
- Pycharm: Integrated development environment used to preprocess the data.

- Python 3.5: Programming language used for the research oTensorflow-GPU: Machine learning framework used to build some of the models. Efficient for matrix operations and GPU usages.
- Keras-GPU: Machine learning Framework with Tensorflow backend. Useful to build neural network and tune hyperparameters.
- OpenCV: Computer vision framework used for preprocessing.

4.4 RESULTS

4.4.1 Input Given

Initially the input is been given as a picture from our system. Here the network is been trained for twelve animals. Any one of the animal's picture is given as input respectively. For example, here the image of an elephant is given as the input image to the network.

Figure 4.2 : Input Image

4.4.2 Processing of Input Image

In this application the image from the camera trap is taken and given as input to find the animal. The Captured image can be of in any time considering the lighting in the image and also the environmental conditions may vary like Summer, Winter, Spring or Autumn. However the captured animal should be identified successfully in order to find what animal is intruding. Considering all these factors the image pre-processing techniques are undergone. The Input image is processed. The Gray Image, HSV image and the Binary Image of the input is obtained by processing the Input.

Figure 4.3 : Gray scale Image

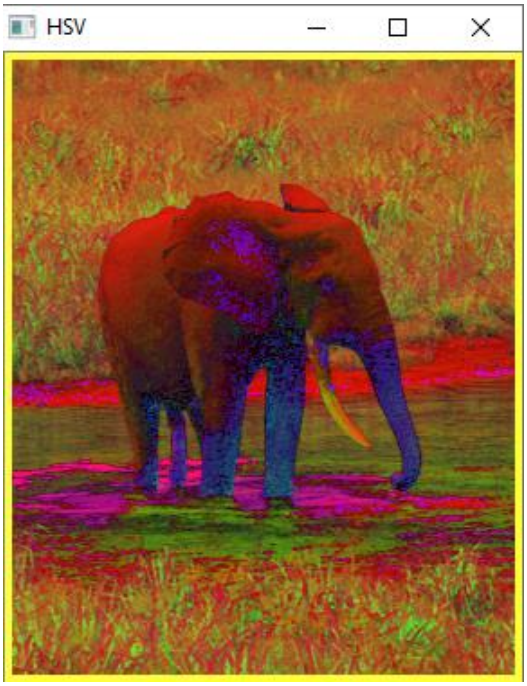


Figure 4.4 : HSV Image

Figure 4.5 : Binary Image

Figure 4.6 : Result Image

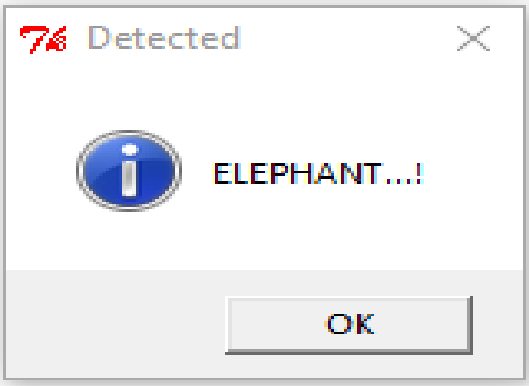


Figure 4.7 : Popup Image

4.4.3 Result

Finally the result is obtained from the program. An alert message will be popping up in the screen. As the Elephant Picture is given as input it will pop-up as Elephant is detected. Following that, the message will also been shown in the python shell window.

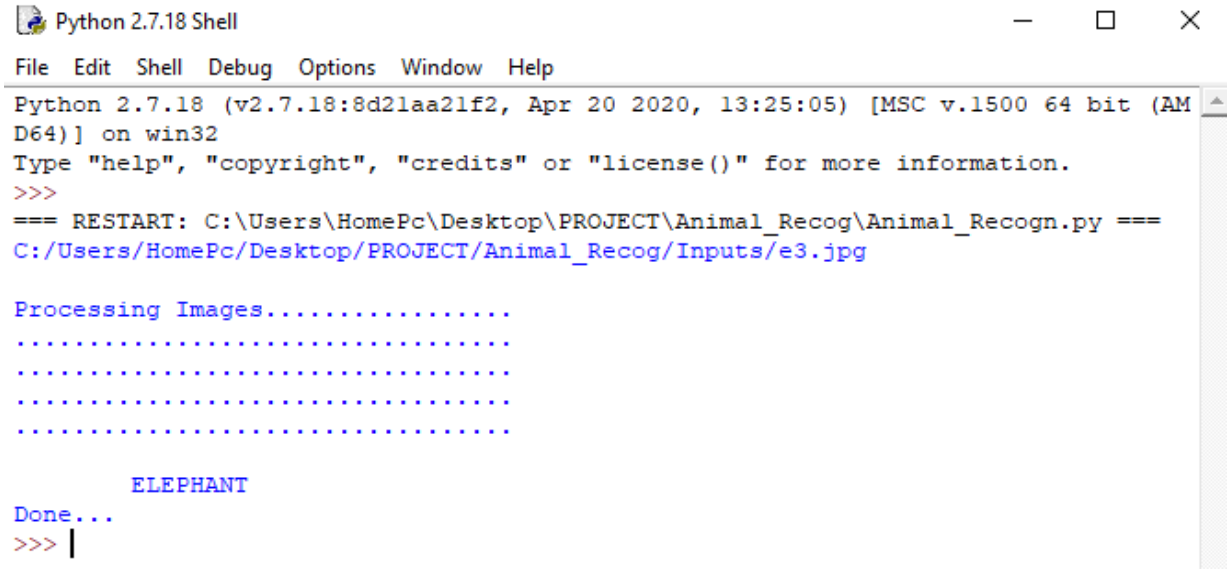


Figure 4.8 : Python 2.7.18 shell

By giving various inputs to the network the accuracy of the network is calculated. The Overall Accuracy we got is 96.48% which is better than the traditional Convolutional Neural Networks respectively. Thus, by working on the Capsule Network Algorithm it is proved that it is a better algorithm than CNN which satisfies the drawbacks of them.

CHAPTER 5

SOFTWARE DESIGN

5.1 PYTHON VERSION

Versioin : 2.7.18

5.1.1 PYTHON

Figure 5.1 PYTHON

Python programming language is a high-level programming language which is widely used for general purpose. It's design philosophy improves the code readability, and the syntax allows Users to express their concept in fewer lines of code than would also be possible in other languages like C++ or Java. For both small and large scale this language provides constructs intended to enable clear programs respectively.

Python programming language supports multiple programming paradigms, which includes object-oriented, imperative and procedural styles or functional programming. It is having a standard large and comprehensive library which are featured with dynamic tape system.

For the installation on many operating systems python interpreters are available, which allows Python code to execute on a wide variety of system. Using third-party tools, such as Py2exe or Pyinstaller,

The Codes can be packaged into a stand-alone executable programs for operating systems which are most popular, and allows for the distribution of Python-based softwar to use on those environments which does not require a python interpreter for installation.

CPython which is a python reference implementation, is a free and open-source software and also it has a comunnity-based development model, as they all of the nearly alternative Implementations. It is been managed by the Python Software Foundation which is a non profit one.

The exact syntax and semantics of the Python language were described by the python language reference, the standard library that is distributed with Python is manually described by the library reference.And also Optional components are described by them which are commonly included in the pyhton distributions.

The Python's standard library offers a wide range of facilities which is very extensive. It provides access to system functionality like file input or output that would otherwise be not accessible to Python users, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming as it contains built-in modules (Written in C program). To encourage and enhance the portability of Python programs by abstracting away platform-specifics into platform-neutral APIs, some of these modules are explicitly designed.

The Installers of python for the Windows platform usually consists of the entire standard library and often also includes many additional components. It may be necessary to use the packaging tools provided with operating system to obtain some or all of the optional components since Unix-like operating systems python will be normally provided as collection of packages. There is a growing collection of several thousand components in addition to the standard library, available from the

python package index for the individual programs and modules to packages and for the development of entire applications and frameworks.

5.2 OPERATING SYSTEM

Windows 10

5.3 ADDITIONAL PACKAGES

Open CV 2.4.10 (CV2)

Pillow

5.3.1 OpenCV -PYTHON

Figure 5.2 OpenCV - PYTHON

OpenCV-Python is a library of Python bindings designed to solve computer vision problems. It is used for all sorts of image and video analysis. It is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structure are converted to and from Numpy arrays.

5.3.2 PILLOW – PYTHON

Figure 5.3 Pillow– PYTHON

Python-pillow.org. Python Imaging Library is a free and open-source additional library for the python programming language that adds support for opening, manipulating and saving many different image file formats. It is available for Windows, Mac, OS X and Linux.

CHAPTER 6

FUTURE WORK AND CONCLUSION

6.1 FUTURE WORK

With the improvements and availability of more resources, the project will explore more areas of this research. How to widen capsule network and the impact it has on complex datasets. Using transfer learning for feature extraction and feed the features to the capsules. Explore new preprocessing techniques that might improve our models. Study the behavior of capsule networks without prior convolutions applied. Finally, Repeat the whole experiment for a complex dataset. The first perspective we plan to explore next is actually the impact of having a bigger or smaller capsule along with the number of capsules per layers. This inspiration is driven by the idea behind Inception networks, however, there is a doubt that is will reduce the computational needs of the network as in Inception.

Our second perspective is to use the well-trained CNNs as feature extractors on high-quality images and then pass the features to the capsules for the whole dataset. This will drastically reduce the computational cost of having a whole capsule network built to receive a 300*300 image. The Third perspective is about optimizing all the previous and aforementioned processes to run them on the complex Dataset.

6.2 CONCLUSION

In this Project , a dataset consisting of twelve animals has been exploredand tried to out-perform existing authors that attempted the same identification task. Moreover, quite complex network in implementation has been explored. For the former, the project succeeded to have our top-1 accuracy up to 96.48%.

As for the later, it has been shown how capsule networks have a big potential in recognizing complex objects in very low-quality images. However, the more depth importance give to the capsule, the higher the computational expenses rise; they

rise exponentially. The Project have also experienced manual and convolutional dimensionality reduction applied to a capsule network. A capsule network learns better from a convolutional cropping. Indeed, it gives it the ability to be built for any size and resolution of images.

More importantly, reaching 96.48% on automatically identifying animals in the wilderness is a great advancement as it performs as good as a human expert on a clear picture and better than human experts on “dark pictures”. Recalling that it took more than 8 years to human experts to classify the SS dataset, it is with a scientific optimism that we believe we can even do better by refining a little more our pre-processing techniques and optimizing our networks. Indeed, one of basic difficulty of this research is the exploratory analysis of the capsule network as it is computationally greedy. The more width and depth you add to the network, whether at the level of capsules or convolutions, the more the required computational power will increase.

The capacities of the capsules seem only limited because of the huge computational needs to run larger networks on a larger dataset. However, we expect that not to be a problem in the coming years as computational power becomes more available to the common public.

REFERENCES

1. Margarita Favorskayaa & Andrey Pakhirkaav(2019), "Animal species recognition in the wildlife based on muzzle and shape features using joint CNN" , Reshetnev Siberian State University of Science and Technology, 31 Krasnoyarsky Rabochy ave., Krasnoyarsk, 660037 Russian Federation, pp. 3-4

2. Xiaoyuan Yu, Jiangping Wang, Roland Kays, Patrick A Jansen, Tianjiang Wang and Thomas Huang (2013), "Automated identification of animal species in camera trap images", EURASIP Journal on Image and Video Processing , pp.2-4.

3. Jhony-Heriberto Giraldo-Zuluaga, Augusto Salazar, Alexander Gomez, Angélica Diaz-Pulido (2017), "Automatic Recognition of Mammal Genera on Camera-Trap Images using Multi-Layer Robust Principal Component Analysis and Mixture Neural Networks", Grupo de Investigación SISTEMIC, Universidad de Antioquia, Medellín, Colombia, Instituto de Investigación de Recursos Biológicos Alexander von Humboldt, Bogotá D.C, Colombia, pp.3-5

4. M.N.Favorskaya & V.V.Buryachenko (2018), "Background extraction method for analysis of natural images captured by camera traps", Reshetnev Siberian State University of Science and Technology, Russian federation, pp.2-7

5. Joel Kamdem Teto & Ying Xie (2018), "Automatic identification of animals in the wild: a comparative study between c-capsule networks and deep convolutional neural networks" , Kennesaw State University, pp.21-28

6. Edgar Xi, Selina Bing & Yang Jin (2017), "Capsule Network Performance on Complex Data", Carnegie Mellon University Pittsburgh, pp.1-3

7. By Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton(2017), "ImageNet Classification with Deep Convolutional Neural Networks", Communications of ICM, pp.86-89

8. Amara Dinesh Kumar, R.K arthika & Dr. Latha Parameswaran (2018), "Novel Deep Learning Model for Traffic Sign Detection Using Capsule Networks", International Journal of Pure and Applied Mathematics, pp.4543-4546

9. Zhenhua Chen & David Crandall (2018), "Generalized capsule networks with trainable routing procedure" , School of School of Informatics, Computing, and Engineering Indiana University Bloomington, pp.2-4

10. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens (2015), "Rethinking the Inception Architecture for Computer Vision", Zbigniew Wojna University College London, pp.2-7