Project Report on

IMAGE BASED SPECIES CLASSIFICATION USING CNN

Submitted by

S.KALYANI BAI-S170831
C.LATHA-R170840

Under the guidance of

P.UDAYASREE

Assistant Professor

Department of Computer Science and Engineering

RK VALLEY

Rajiv Gandhi University of Knowledge and Technologies (RGUKT),R.K.Valley ,Kadapa ,Andhra Pradesh.

DECLARATION

We, hereby declare that this report entitled “IMAGE BASED SPECIES CLASSIFICATION USING CNN” submitted by us under the guidance and supervision of P.UDAYASREE is a bonafide work . We also declare that it has not been submitted previously in part or in full to this university or other university or institution for the award of any degree or diploma. All information included from other sources have been duly acknowledged.

We will be solely responsible if any kind of plagiarism is found.

Place : R.K.Valley

S.KALYANI BAI-S170831
C.LATHA-R170840

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts success.

I am extremely grateful to our Director.Prof. K. SANDHYA RANI for fostering an excellent academic climate in our institution.

I also express my sincere gratitude to our respected Head of the Department Mr.N.SATYANANDARAM for his encouragement, overall guidance in viewing this project a good asset and effort in bringing out this project.

I would like to convey thanks to our guide at college Ms.P.UDAYASREE for her guidance, encouragement, co-operation and kindness during the entire duration of the course and academics.

My sincere thanks to all the members who helped me directly and indirectly in the completion of project work. I express my profound gratitude to all our friends and family members for their encouragement.

INDEX

Abstract:-

Animal species image classification is used in forests to classifiy animals in real time. In past, many computer vision techniques were introduced but they couldn't fulfill the requirements as the accuracy got depreciated since the technology advanced. But as per the requirement many techniques were introduced where accuracy got drastically improved where we could perform the image classification, image recognition and segmentation. This project aims to introduces efficient technique for animal species image classification with the goal of achieving good amount of accuracy. Convolutional neural network is been engineered for the image classification process. Bottleneck features are trained and synched to the pretrained architecture to achieve high accuracy. Numerous deep learning architectures are compared with the dataset. This goal can be translated into an image classification problem for deep learning models. So I explored a simple neural network, and then progressed to convolutional neural network and transfer learning. And Can automatically help identify animals in the wild taken by wildlife conservatories. Can lead to discoveries of potential new habitat as well as new unseen species of animals within the same class by developing the bottleneck features of an animal based on that it will classify by prediction.

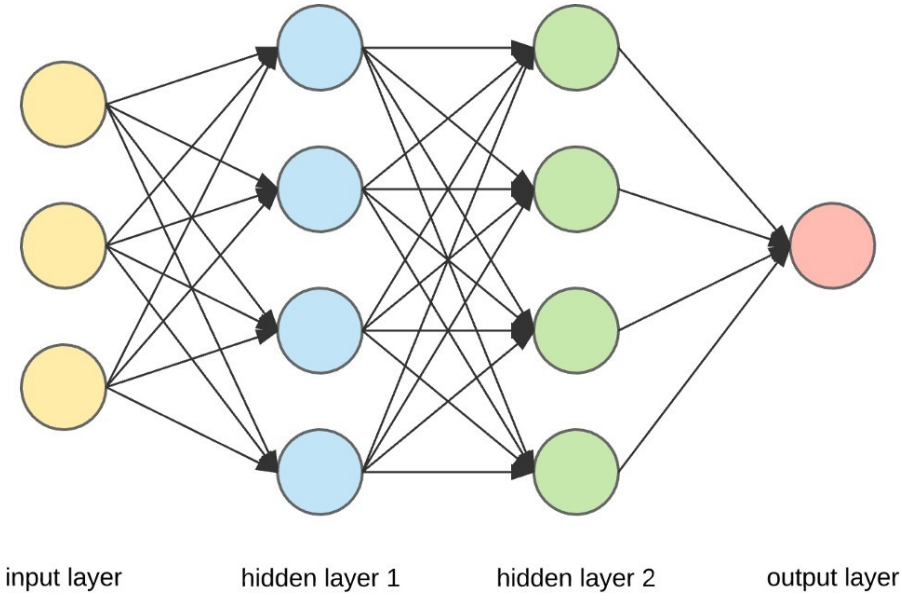
Introduction :-

Today, with the increasing volatility, necessity, and applications of artificial intelligence, fields like neural networks, and its subsets, machine learning, and deep learning have gained immense momentum. It has become a data-centric model where neural network developers are "training" the network to be "intelligent" and "independent". The training needs software and tools such as classifiers, which feed huge amounts of data, analyze them, and extract useful features. These features are then used to observe a pattern and train the network to use similar data again the next time it is fed data. Convolutional Neural Network remains to be the most sought-after choice for computer scientists for image recognition, processing, and classification. We propose a fauna image classifier using a convolutional neural network, which will be used to classify images of different species and animals captured in dense forest environments to achieve the desired accuracy, and aid ecologists and researchers to further study and/or improve habitat, environmental and extinction patterns. A convolutional neural network is trained and developed for efficiently classifying these images with accurate results. Images captured in a field represent a challenging task while classifying since they appear in a different pose, background clutter, different illumination, and climatic conditions, human photographic errors can cause significant amounts of

distortion, different angles, and occlusions. All these challenges necessitate an efficient algorithm for classification with the most optimum accuracy. One of the biggest issues is “class imbalance”. Since there are an uneven number of pictures for each sample, the algorithm could train some categories better, as compared to the others. The scope of the project is to train a neural network for animal image datasets ,which need more memory allocation, good resolution,consistency attributes,and good classification .

What is Neural network?

A series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria. The concept of neural networks, which has its roots in artificial intelligence is swiftly gaining popularity in the development of trading systems. Neural networks are multi-layered networks of neurons. Below is the diagram of a simple neural network with five inputs, 5 outputs, and two hidden layers of neurons.



What is convolutional neural network?

A Convolutional Neural Network, also known as CNN or ConvNet, is a class of neural networks that specializes in processing data that has a grid-like topology, such as an image. A digital image is a binary representation of visual data. It contains a series of pixels arranged in a grid-like fashion that contains pixel values to denote how bright and what colour each pixel should be. The human brain processes a huge amount of information the second we see an image. Each neuron works in its own receptive field and is connected to other neurons in a way that they cover the entire visual field. Just as each neuron responds to stimuli only in the restricted region of the visual field called the receptive field in the biological vision system, each neuron in a CNN processes data only in its receptive field as well. The layers are arranged in such a way so that they detect simpler patterns first (lines, curves, etc.) and more complex patterns (faces, objects, etc.) further along. By using a CNN, one can enable sight to computers.Neural networks accept an input image/feature vector (one input node for each entry) and transform it through a series of hidden layers, commonly using nonlinear activation functions. Each hidden layer is also made up of a set of neurons, where each neuron is *fully connected* to all neurons in the previous layer. The last layer of a neural network (i.e., the “output layer”) is also fully connected and represents the final output classifications of the network.

Layer types :-

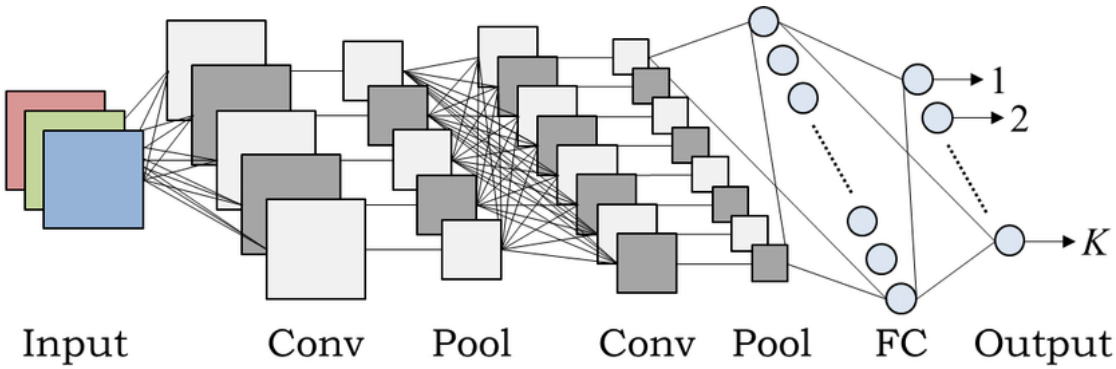
There are many types of layers to build Convolutional neural networks ,but the ones you are most likely to encounter include:

- Convolutional(conv)
- Activation(ACT or RELU)
- Pooling

The majority of computations happen in the convolutional layer, which is the core building block of a CNN. A second convolutional layer can follow the initial convolutional layer. The process of convolution involves a kernel or filter inside this layer moving across the receptive fields of the image, checking if a feature is present in the image.

Over multiple iterations, the kernel sweeps over the entire image. After each iteration a dot product is calculated between the input pixels and the filter. The final output from the series of dots is known as a feature map or convolved feature. Ultimately, the image is converted into numerical

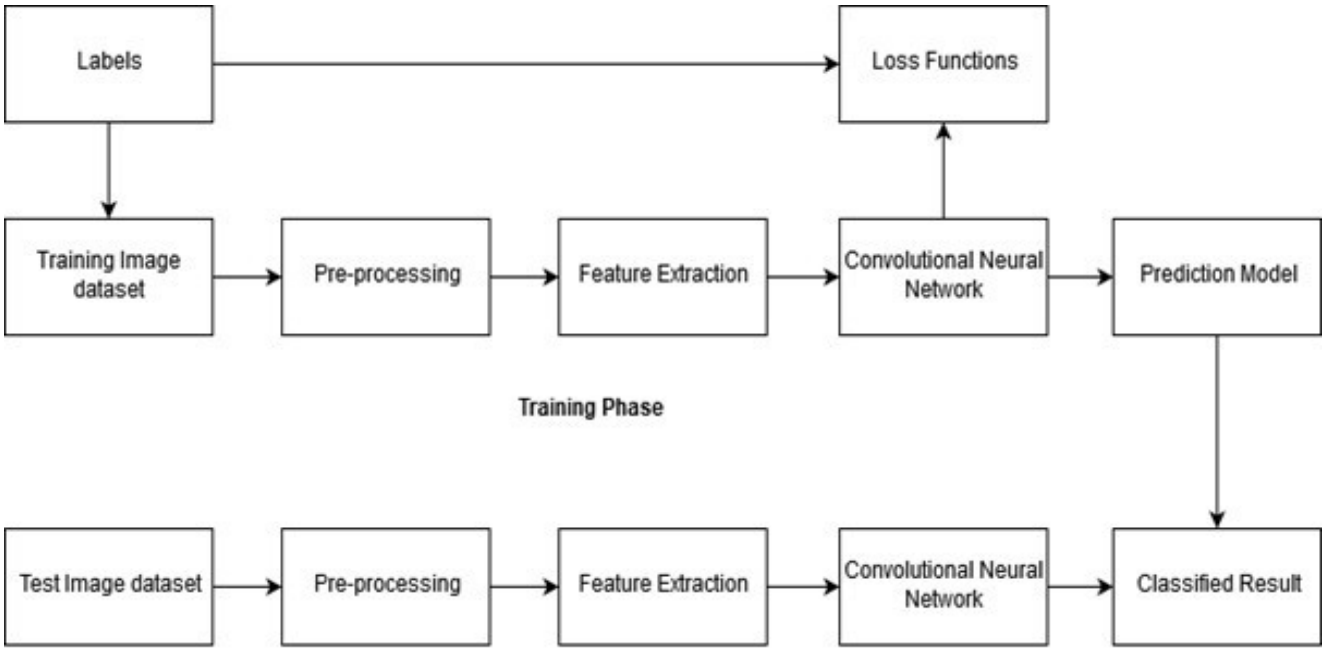
values in this layer, which allows the CNN to interpret the image and extract relevant patterns from it.



Proposed System: -

We have used CNN for this project because image classification involves the extraction of features from the image to observe some patterns in the dataset. Using an ANN for the purpose of image classification would end up being very costly in terms of computation since the trainable parameters become extremely large. if we have a 50 X 50 image of a cat, and we want to train our traditional ANN on that image to classify it into a dog or a cat the trainable parameters become (50*50) * 100 image pixels multiplied by hidden layer + 100 bias + 2 * 100 output neurons + 2 bias = 2,50,302 We use filters when using CNNs. Filters exist of many different types according to their purpose.

Block Diagram of CNN:



5. Requirement Specification

Hardware Configuration:

1.Processor : A processor with a speed of 2.5Ghz or higher is sufficient for this project.

2.Memory: A minimum of 8GB of RAM is recommended.

3.Storage: 500GB of storage space is sufficient for this project.

Requirement Analysis:-

Computer Aided learning is a rapidly growing dynamic area of research in wild life industry. The recent researchers in machine learning promise the improved accuracy of perception of animal classification. Here the computers are enabled to think by developing intelligence by learning. There are many types of Machine Learning Techniques and which are used to classify the data sets.

Dataset :-

We collected the dataset from the Internet .

We have collected two different species,we used dogs and cats in this project.

We splitted the dataset into three different sets.

1.Training set

Which contains the dataset for training the model consists of 4000 images of cats and 4000 images of dogs with pixels(64x64) with 3 channels(rgb).

2.Test set

Which contains the dataset for testing the model after training it, it consists of 1000 images of cats and 1000 images of dogs.

3.Prediction set

For predicting the images we separated 10images from the dataset for output prediction.

Software Analysis:-

The project primarily focuses on sign language detection. We implemented with Python 3.7 version. The libraries required are to install prior to execute the project. We installed CV2 for OpenCV, Keras, TensorFlow, Numpy, etc.

Training - Phase

Pre-processing:-

Images — Channels and Sizes

For example, some images are what we call “natural images”, which means they are taken in **colour**, in the **real world**. For example Images come in different shapes and sizes**.** They also come through:

- A picture of a flower is a natural image.
- An X-ray image is *not* a natural image.
- Taking all these variations into consideration, we need to perform some pre-processing on any image data. RGB is the most popular encoding format, and most “natural images” we encounter are in RGB. Also, among the first step of data pre-processing is **to make the images of the same size.** Let’s move on to how we can **change the shape and form of images.**

Feature Extraction:-

This section gives the reader an intuitive idea of how convolving an image with a kernel will result in extraction of features from the input image. We consider an image of digit 2 that is being convolved with the same 3*3 filter flipped by 90 degrees(which is our choice) in each example.and also reshaping the image for feature extractions.

Dataset:-

we have collected the dataset from the internet.

Tools Used:

Jupyterlab/Googlecolab

Need To import libraries:-

- Matplotlib.pyplot
- Numpy
- Pandas
- Keras
- Tensorflow

KERAS AND TENSORFLOW:

Keras is a minimalist Python library for deep learning that can run on top of Theano or TensorFlow. It was developed to make implementing deep learning models as fast and easy as possible for research and development. It runs on Python 2.7 or 3.5 and can seamlessly execute on GPUs and CPUs given

the underlying frameworks. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. Keras was developed and maintained by François Chollet, a Google engineer using four guiding principles.

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications. TensorFlow is a free and open source software for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. TensorFlow is Google Brain's secondgeneration system. While the reference implementation runs on single devices, TensorFlow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for generalpurpose computing on graphics processing units). TensorFlow is available on 64- bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS. Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers

to mobile and edge devices. TensorFlow computations are expressed as stateful dataflow graphs. The name TensorFlow derives from the operations that such neural networks perform on multidimensional data arrays, which are referred to as tensors.

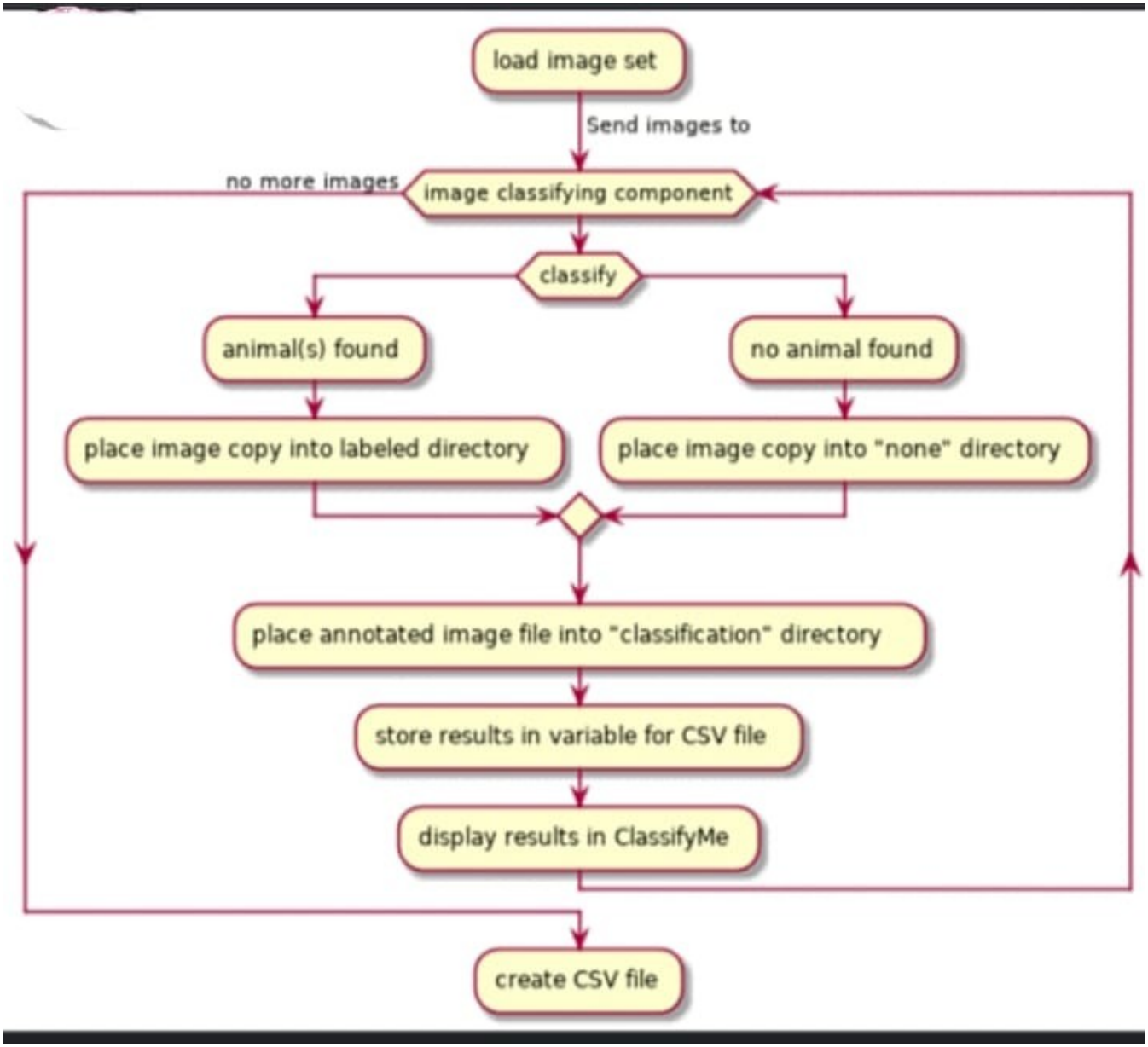
Matplotlib: It is a library for creating static, animated, and interactive visualizations in Python. It provides a wide range of plotting functionality, including line plots, scatter plots, bar charts, histograms, and more.Matplotlib is often used for visualizing data during the exploratory data analysis (EDA) phase of machine learning projects.

NumPy: It is a library that provides support for large, multi-dimensional arrays and matrices, as well as a variety of mathematical functions to manipulate them. It is a fundamental library for scientific computing in Python and is often used for tasks such as data manipulation, linear algebra, and statistical analysis.

Pandas: A library for data manipulation and analysis that provides powerful data structures for working with structured data.

UML Diagrams:

The Unified Modelling Language (UML) is a graphical language for visualizing, specifying, constructing and documenting of a software intensive system. The UML gives a standard way to write a system blueprints, covering conceptual things, such as classes written in a specified programmed language, database schemas and reusable software components.



SYSTEM IMPLEMENTATION

Design Considerations:-

Dataset is large enough as possible. Previous results of CNN framework for animal classifications is taken into consideration.

Description of the Proposed Algorithm:-

The aim of the proposed algorithm is to compare the overall recognition accuracy of animals by using two deep learning neural network frameworks – CNN and Faster RCNN and highlight the importance of deep learning technology in classification problems.

The proposed algorithm is consists of five main steps

Input Image:-

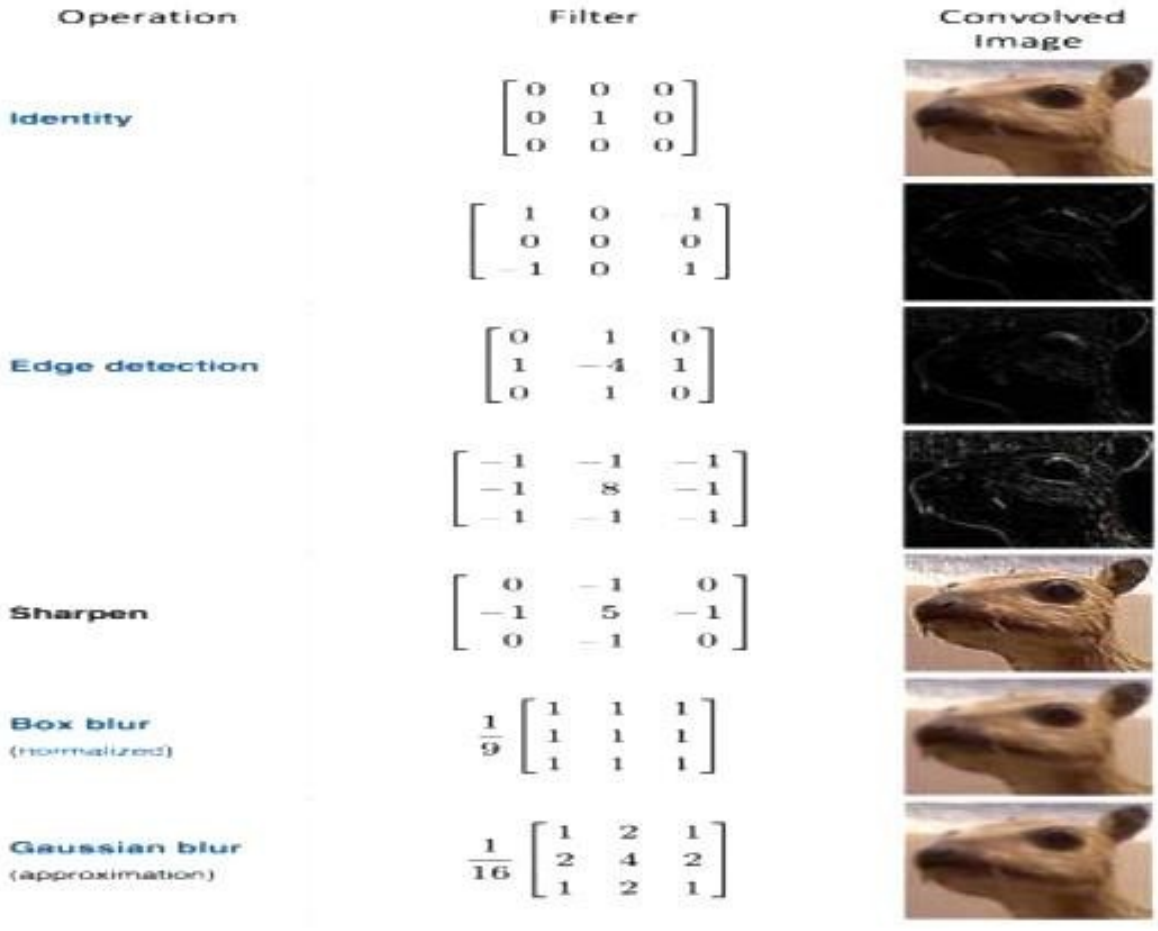
The image uploaded by the user is considered as a input image (test image) and is sent to the network framework. The image is rescaled to a fixed size.

Design layer:-

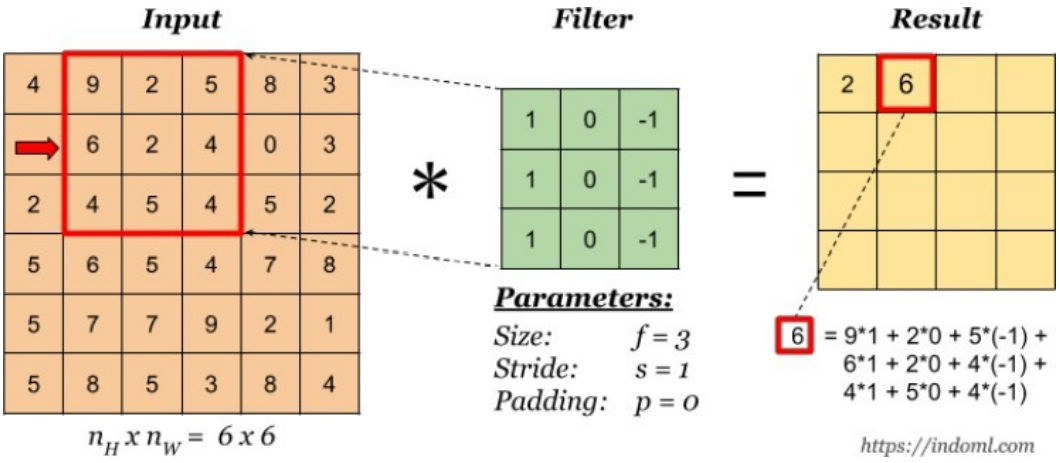
- Convolutional layer
- Rectified linear unit layer
- Pooling layer

Convolutional layer:-

The Convolutional layer is can be considered as a building block of a Convolutional Neural Network as it does most of the computational heavy lifting. The network extracts and learns most of the features in this layer. The image is represented as a matrix. A filter that is used as a feature matrix is slide along the image matrix to obtain the convolved feature map. Many different operations can be performed to extract different features by changing the values in the filter matrix.



The basic convolution operation can be depicted below: Overlay the filter to the input, perform element wise multiplication, and add the result. Move the overlay right one position and do the same calculation above to get the next result.



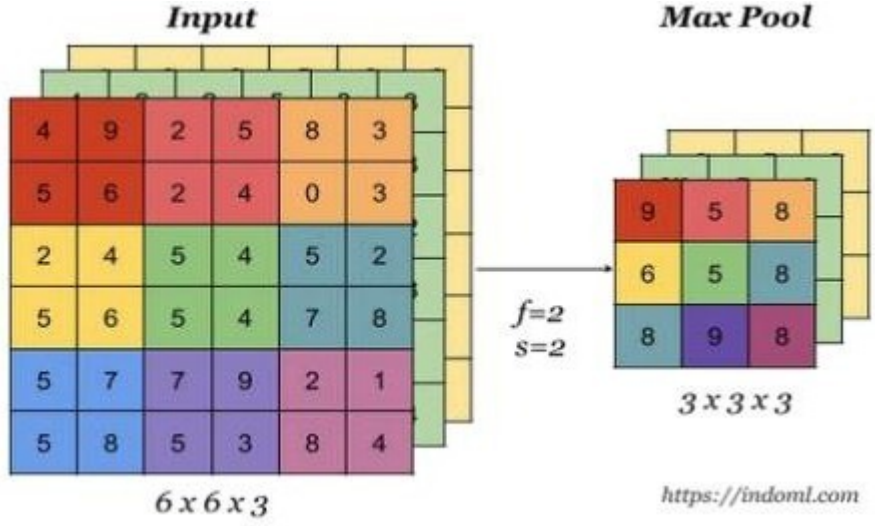
Stride tells us how many cells the filter is to be moved in the input to calculate the next cell in the result. Padding uses a convolutional layer without shrinking the height and width of the volumes. This is important for building deeper networks, since otherwise the height or width would shrink as we go to deeper layers.

Pooling layer:-

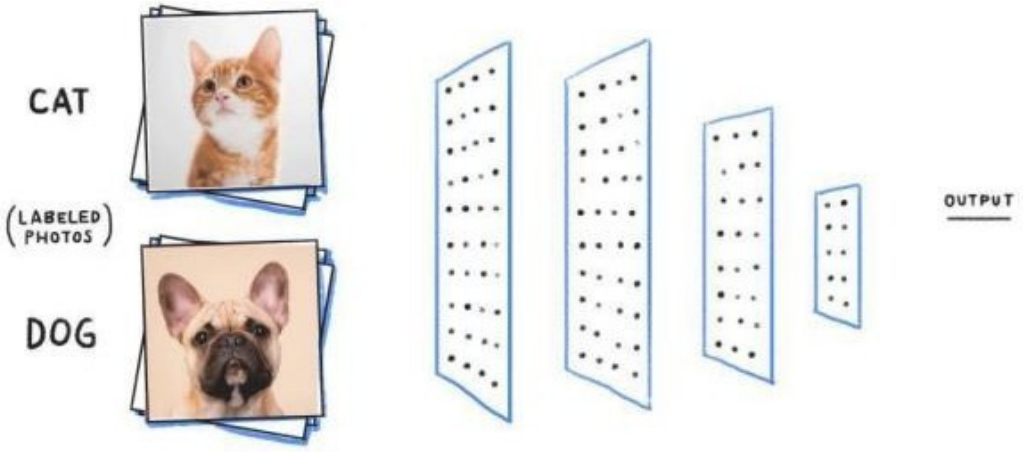
The pooling operation provides spatial variance making the system capable of recognizing an object with some varied appearance. It involves adding a 2Dfilter over each channel of the feature map and thus summarise features lying in that region covered by the filter.

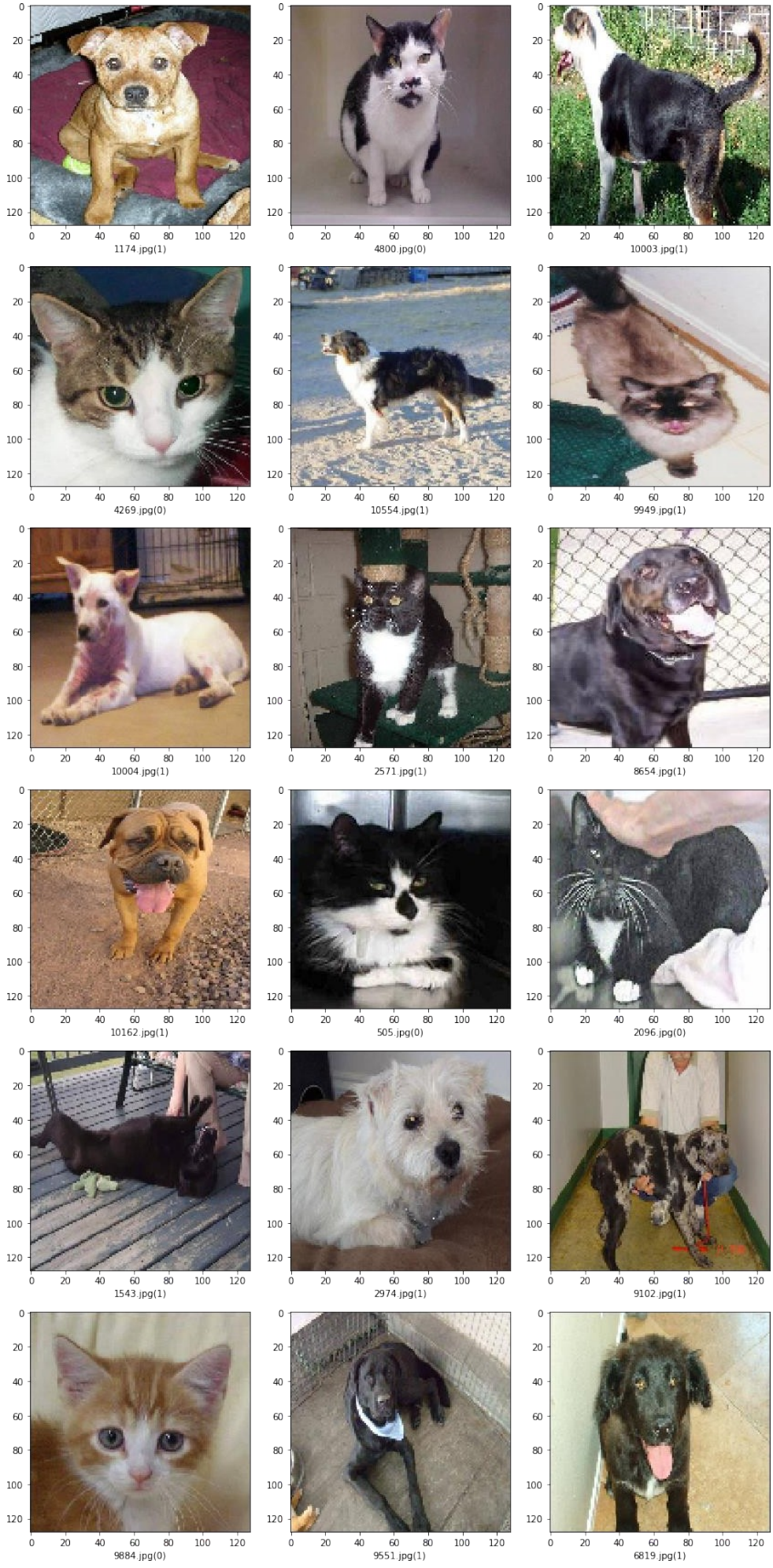
So, pooling basically helps reduce the number of parameters and computations present in the network. It progressively reduces the spatial size of the network and thus controls overfitting. There are two types of operations in this layer; Average pooling and Maximum pooling. Here, we are using max-pooling which according to its name will only take out the maximum from a pool. This is possible with the help of filters sliding through the input and at each stride, the maximum parameter will be taken out and the rest will be dropped.

The dimensionality of the feature map is reduced to get shrinker maps that reduces the parameters, weights and computations. Pooling can be Max, Average or Sum Pooling from the rectified and downsized feature map. Number of filters in convolution layer will be same as the number of output maps from pooling. It makes the network invariant to transformations, distortions and translations in the input image.

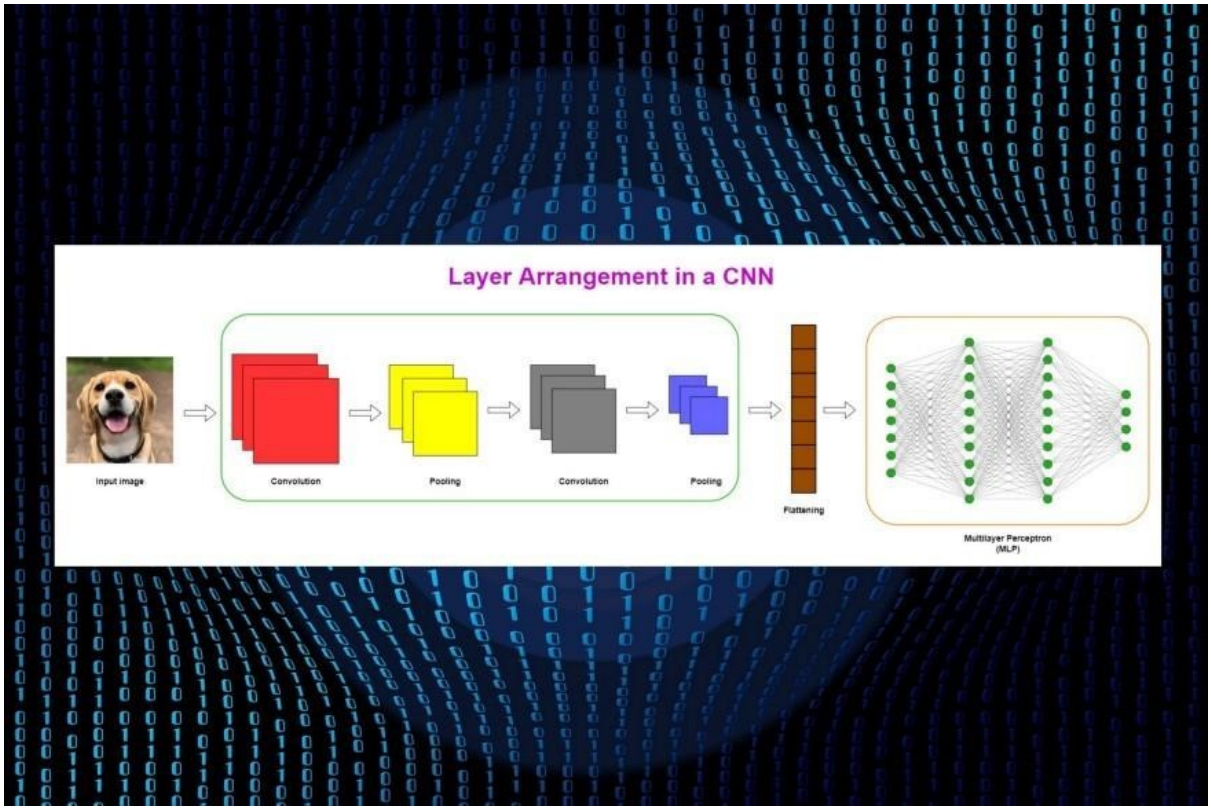


Species used for classification:-




1174.jpg(1)
4800.jpg(0)
10003.jpg(1)
4269.jpg(0)
10554.jpg(1)
9949.jpg(1)
10004.jpg(1)
2571.jpg(1)
8654.jpg(1)
10162.jpg(1)
505.jpg(0)
2096.jpg(0)
1543.jpg(1)
2974.jpg(1)
9102.jpg(1)
9884.jpg(0)
9551.jpg(1)
6819.jpg(1)

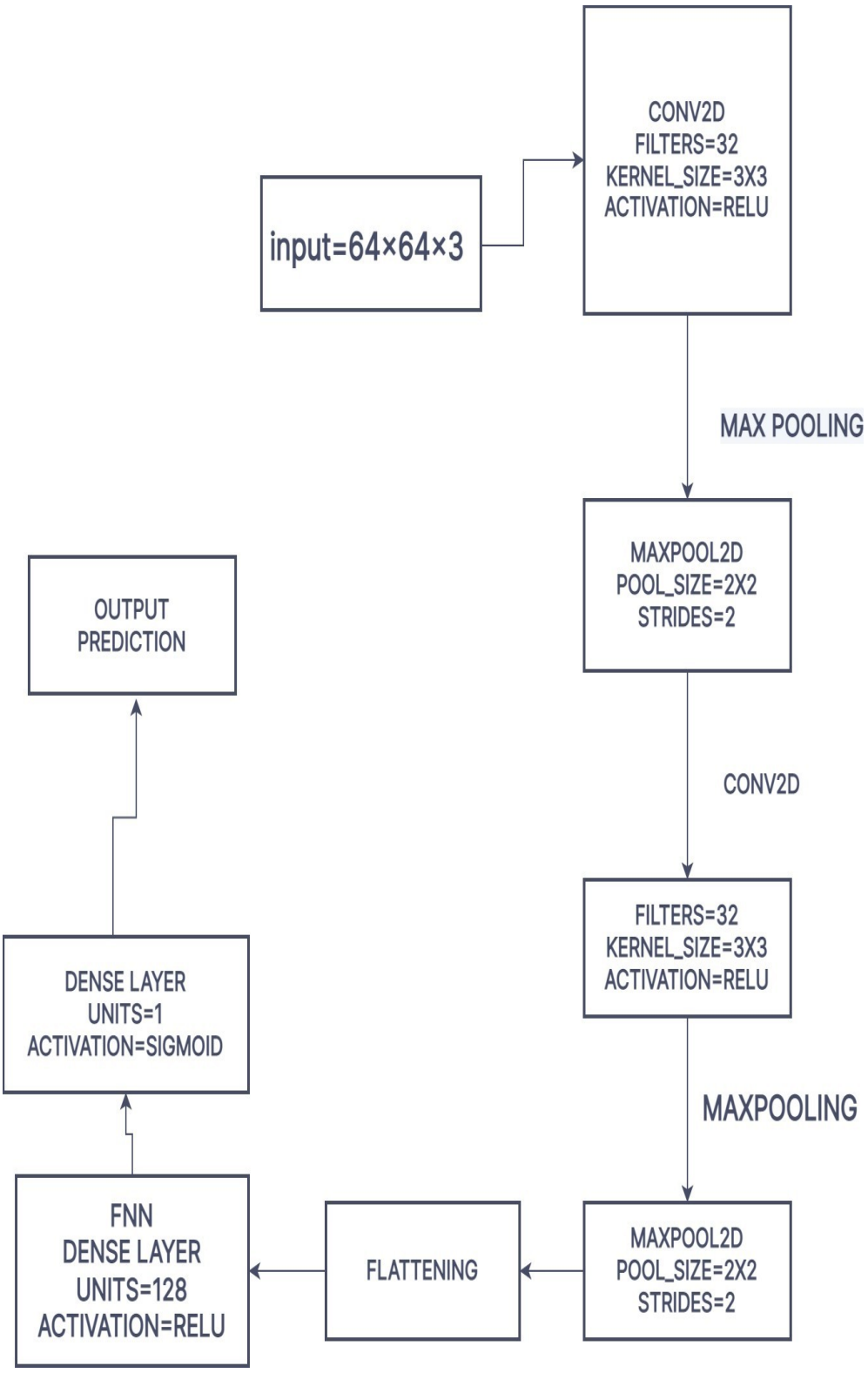
Layer arrangement in CNN:-



Each convolutional layer in a CNN is created usi the Conv2D()class that simply performs the convolution operation in a two-dimensional space. In other words, the movement of the kernel (filter) happens on the input image across a two-dimensional space.

In keras a convolutional layer is referred to as a Conv2D layer.

Architecture :
-



Code:

```
pip install tensorflow  pip install keras  import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator  train_datagen
= ImageDataGenerator(rescale = 1./255,
                                shear_range = 0.2,
                                zoom_range = 0.2,  horizontal_flip =
                                True)
training_set =
train_datagen.flow_from_directory('datasets/dogs_cats/training_set',
                                                target_size = (64, 64),
                                                batch_size = 32,
                                                class_mode = 'binary')

test_datagen = ImageDataGenerator(rescale = 1./255)
test_set =
test_datagen.flow_from_directory('datasets/dogs_cats/test_set',
                                            target_size = (64, 64),
                                            batch_size = 32,  class_mode =
                                            'binary')
cnn =  tf.keras.models.Sequential()
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3,
activation='relu',  input_shape=[64, 64, 3]))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3,
activation='relu'))  cnn.add(tf.keras.layers.MaxPool2D(pool_size=2,
strides=2))  cnn.add(tf.keras.layers.Flatten())
cnn.add(tf.keras.layers.Dense(units=128, activation='relu'))
cnn.add(tf.keras.layers.Dense(units=1, activation='sigmoid'))
cnn.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics =
 ['accuracy'])
cnn.fit(x = training_set, validation_data = test_set, epochs = 25)
import numpy as np
from keras.preprocessing import image
test_image =
image.load_img('datasets/dogs_cats/single_prediction/cat_dog11.jpg',
target_size = (64, 64))
test_image = image.img_to_array(test_image)  test_image =
np.expand_dims(test_image, axis = 0)  result = cnn.predict(test_image)
training_set.class_indices
if result[0][0] == 1:  prediction = 'dog'
else:
prediction = 'cat'
```

```
print(prediction)
```

Accuracy:-

We achieved the 90.84% of accuracy.

Advantages :-

1.This neural network accepts two dimensional array data as input and will give a number as output corresponding to the class.

2.There is no transfer learning does happened in machine learning algorithms so in deep learning there is transfer learning happens then it will learn more from the dataset and also less error will occurred.

3.And it also automatically extracted features from the Image.

4.CNN uses adjacent pixel information to effectively downsample the image first by convolution and then uses a predictor layer at the end.

5.It gives better accuracy than ANN and for 3 channel images CNN gives best accuracy.

Disadvantages :-

6.They take a lot of data(for more than other algorithms) to train.

7.they take a lot of time(for more than other algorithm) to train.

3.The huge disadvantage of multi-layered models such as the CNNs is that on each layer,different level of visual information is processed.Lower layers process very local features just small parts of curves etc.the higher you get the more complex features are concerned.

4.This projects works better in python 3.7/3.8 may vary for other versions.

Conclusion :-

The architecture and parameter used in this network are capable of producing accuracy of 90.84% on validation Data which is pretty good.It is possible to Achieve more accuracy on this dataset using deeper network and fine tuning of network parameters(filters,epochs etc) for training.

REFRENCES:

https://www.researchgate.net/publication/327000784_Machine_learning_for_image_based_species_identification

https://docs.python.org/3/library/index.html