International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 11 | November - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Convolution Neural Network- an Advanced Deep Learning Algorithm for Image Classification and Performance Evaluation

Akshatha S.A

Computer Science and Engineering,

Bearys Institute of Technology, Mangalore

Email address: 69aksa@gmail.com (Akshatha S.A)

---***---

Abstract - Artificial Intelligence is frequently described as an intelligent system equipped with the capacity for logical thinking and problem-solving, akin to a human being, but without genuine consciousness. The origin of AI dates back 8,000 to 11,000 years, with stories of artificial beings endowed with consciousness by ancient rishis of India. Machine Learning is often defined as "the capacity to acquire knowledge without the need for explicit programming." Deep Learning is part of ML consisting of algorithms that allow software to train itself to do the complex tasks such as speech recognition and image recognition, among others. An artificial neural network is an interconnected group of nodes that mimics the function of neurons in the brain. Deep learning employs artificial neural networks to conduct intricate computations on extensive datasets. Deep learning supports Convolutional Neural Networks (CNNs) to recognize and classify images. CNNs are a form of supervised learning, reliant on labeled data for training neural networks to identify and categorize images. In the present study, a Convolutional Neural Network algorithm was employed on a specific dataset, and its performance was assessed. The implementation followed a sequential approach within a Jupyter Notebook. The objective of the Convolutional Neural Network models was to discern between cats and dogs, involving two distinct classes. Ultimately, the model achieved a dataset accuracy of 81.25%.

***Key Words*: Convolution Neural Network, Artificial Intelligence, sequential model, Convolution layer, Image detection, dataset.**

1. INTRODUCTION

Convolution Neural Network is an algorithm of Deep Learning that can take in an input image, assign learnable weights and biases to various objects in the image, and be able to differentiate one from the other. CNN falls under the supervised learning category of neural networks which uses labeled input data to identify the given image. The primary objective of the research is to categorize 500 collective images into two groups: cats and dogs.

Rahul Chauhan et al. [1] worked with two different datasets: MNIST and CIFAR-10. The MNIST dataset consists of handwritten digits and is used to evaluate the performance of a classification algorithm. The primary objective of the research is handwritten digit recognition using MNSIT dataset. The CNN model applied to the MNIST dataset attained 99.6% accuracy after 10 epochs of training. Another dataset used in this study is CIFAR-10, which is designed for object detection tasks. It classifies objects into 10 distinct classes and is employed to detect objects within the test sets. The CNN model, when applied to the CIFAR-10 dataset, achieved an accuracy of 80.17%.

Atul Sharmaa et al. [2] presented an image classification approach in their paper. They classified the images into 3 categories: car, aeroplane and bird. They created CNN testing model in python notebook to classify the images. They got an accuracy of 94.2% by using CNN.

Muthukrishnan Ramprasath et al. [3] tested the MNSIT dataset for image classification. This dataset serves a dual purpose, being utilized for both training and testing by the CNN. The resulting accuracy rate is 98%. The training set comprises small, grayscale images.

Md. Anwar Hossain et al. [4] tested the CIFAR-10 dataset with 10,118 test cases, of which a total of 661 images were misclassified. After 300 epochs, the model achieved a testing accuracy of 93.47%. The model took nearly 3 hours to train using a simple algorithm for the CNN model. The designed CNN model is capable of recognizing objects in blurry images. Some missed pixels in the data images resulted in the system's inability to identify certain images.

P. Lakshmi Prasanna et al. [5] conducted an evaluation on the CIFAR-10 dataset, which comprises 60,000 training images with dimensions of 32*32, categorized into ten classes, including airplanes, cats, cars, deer, birds, frogs, dogs, ships, horses, and trucks.

In their research, these authors have achieved great accuracy using CNN. Therefore, we can conclude that CNN is efficient in processing vast amounts of data and producing highly accurate predictions.

This article focuses on the implementation of Convolutional Neural Network models for image recognition, specifically using the Kaggle Cats and Dogs dataset. The article delves into the deployment of these models and assesses their performance, particularly in relation to accuracy.

This document shows the suggested format and appearance of a manuscript prepared for SPIE journals. Accepted papers will be professionally typeset. This template is intended to be a tool to improve manuscript clarity for the reviewers. The final layout of the typeset paper will not match this template layout.

2. Experimental Method

A. Data Sets:

The Kaggle Cats and Dogs dataset is a widely used computer vision dataset designed for the classification of images as either cats or dogs. This dataset comprises a total of 12,000 images, of which we have specifically selected 500 cat images and 500 dog images for our analysis. Below is Figure 1, featuring sample images from the Kaggle Cats and Dogs dataset.

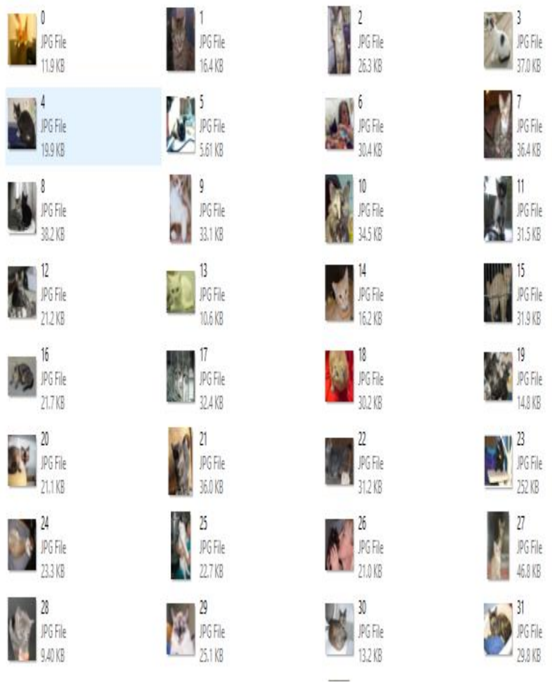

Figure 1: Images of cats from Kaggle cats and Dogs dataset.

B. CNN Models:

Convolutional Neural Network (CNN) is a type of deep learning model especially designed to process grid data, such as images and videos.

CNN theory involves the following:

1. Convolution Operation
2. Convolutional Layer
3. Pooling Layers
4. Activation Functions
5. Fully Connected Layers
6. Training and Backpropagation
7. Dropout and Regularization
8. Transfer Learning

In the research Convolutional neural network model has architecture as follows:

Conv2D -> relu -> MaxPooling2D
Conv2D -> relu -> MaxPooling2D
Conv2D->relu->MaxPooling2D
Flatten layer -> fully connected layer 1 -> fully
Connected layer 2

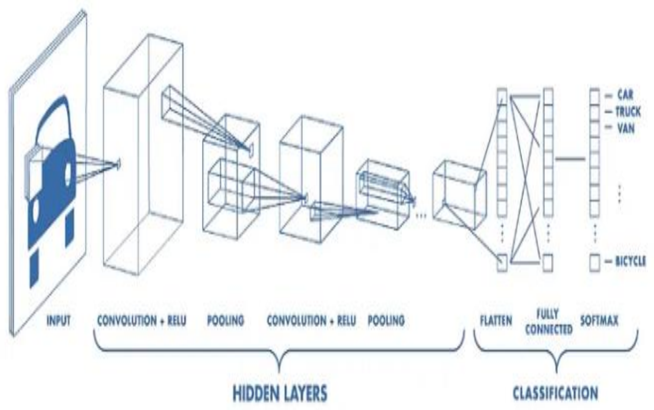

Fig - 2: Architecture model design.

Input shape is (200,200,3) indicating that the input images are 200x200 pixels with 3 color channels (RGB).16 filters in convolution layer 1 is of kernel size 3x3, 32 filters in convolution layer 2 is of kernel size 3x3 and 64 filters in convolution layer 3 of kernel size 3x3. The "same" padding in each convolution layer ensures that the output has the same spatial dimensions as the input.

The CNN model for Kaggle Cats and Dogs dataset is as follows:

Batch size (test): 6,

Number of epochs: 10

Dropout: No

Optimizer: RMS prop, Learning rate=0.001

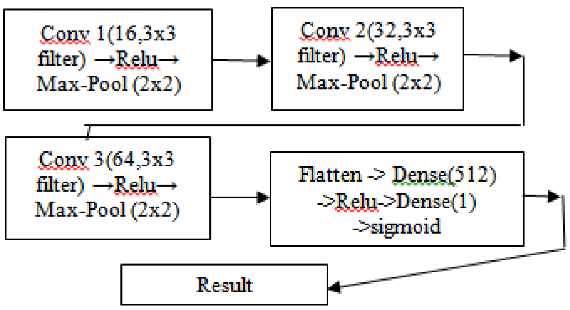

Fig - 3: CNN model for Kaggle Cats and Dogs dataset.

The diagram depicts the architectural design of the model. Here, we create a sequential model by passing a list of layers to the Sequential constructor. We can also create a sequential model incrementally via the add() method and remove layers from the sequential model using the pop() method. The first convolutional layer captures 32 features with a 3x3 filter and uses the "relu" activation function, while the max-pooling layer computes the ultimate value for each patch of the feature map. The second convolutional layer captures 32 features with the same 3x3 filter, uses the "relu" activation function, and is followed by another max-pooling layer. The last convolutional layer captures 64 features with the same 3x3 filter, uses the "relu" activation function, and is followed by max-pooling. The output is subsequently flattened and propagated through two dense layers to yield the ultimate output. This particular model attains an accuracy rate of 81.25 percent.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 200, 200, 16)      448

 max_pooling2d (MaxPooling2D  (None, 100, 100, 16)     0
 )

 conv2d_1 (Conv2D)           (None, 100, 100, 32)      4640

 max_pooling2d_1 (MaxPooling  (None, 50, 50, 32)       0
 2D)

 conv2d_2 (Conv2D)           (None, 50, 50, 64)        18496

 max_pooling2d_2 (MaxPooling  (None, 25, 25, 64)       0
 2D)

 flatten (Flatten)           (None, 40000)             0

 dense (Dense)               (None, 512)               20480512

 dense_1 (Dense)             (None, 1)                 513

=================================================================
Total params: 20,504,609
Trainable params: 20,504,609
Non-trainable params: 0
_________________________________________________________________
```

Fig - 4: summary of sequential model.

The summary() method of the Sequential() class provides a useful output summary containing information about the neural network architecture. The first column displays all the layers that have been created, and the second column shows the output shape of each corresponding layer.

3. Result

Here are the experiment results:

Accuracy: 81.25%, as depicted in Figure 5.

As the number of epochs increases, the model begins to exhibit greater accuracy. Eventually, the model reaches an accuracy of 81.25% at the end.

```
Epoch 1/10
32/32 [==============================] - 27s 812ms/step - loss: 1.2020 - accuracy: 0.5259 - val_loss: 0.6712 - val_accuracy: 0.
6875
Epoch 2/10
32/32 [==============================] - 24s 764ms/step - loss: 0.6877 - accuracy: 0.5719 - val_loss: 0.6687 - val_accuracy: 0.
5000
Epoch 3/10
32/32 [==============================] - 24s 734ms/step - loss: 0.6737 - accuracy: 0.5888 - val_loss: 0.6303 - val_accuracy: 0.
7500
Epoch 4/10
32/32 [==============================] - 25s 780ms/step - loss: 0.6613 - accuracy: 0.6148 - val_loss: 0.6069 - val_accuracy: 0.
5625
Epoch 5/10
32/32 [==============================] - 24s 727ms/step - loss: 0.6186 - accuracy: 0.6637 - val_loss: 0.6715 - val_accuracy: 0.
5625
Epoch 6/10
32/32 [==============================] - 23s 716ms/step - loss: 0.5507 - accuracy: 0.7156 - val_loss: 0.8738 - val_accuracy: 0.
4375
Epoch 7/10
32/32 [==============================] - 23s 719ms/step - loss: 0.5237 - accuracy: 0.7405 - val_loss: 0.5984 - val_accuracy: 0.
6250
Epoch 8/10
32/32 [==============================] - 25s 763ms/step - loss: 0.4702 - accuracy: 0.7814 - val_loss: 0.3992 - val_accuracy: 0.
7500
Epoch 9/10
32/32 [==============================] - 22s 690ms/step - loss: 0.4221 - accuracy: 0.8253 - val_loss: 0.4641 - val_accuracy: 0.
6875
Epoch 10/10
32/32 [==============================] - 22s 691ms/step - loss: 0.3485 - accuracy: 0.8493 - val_loss: 0.4531 - val_accuracy: 0.
8125
```

Fig- 5: Experimental result I.

```
cat1.PNG: dog
1/1 [==============================] - 0s 24ms/step
cat2.PNG: dog
1/1 [==============================] - 0s 33ms/step
cat3.PNG: cat
1/1 [==============================] - 0s 27ms/step
cat4.PNG: cat
1/1 [==============================] - 0s 23ms/step
cat5.PNG: cat
1/1 [==============================] - 0s 19ms/step
cat6.PNG: cat
1/1 [==============================] - 0s 37ms/step
dog1.PNG: dog
1/1 [==============================] - 0s 39ms/step
dog2.PNG: dog
1/1 [==============================] - 0s 18ms/step
dog3.PNG: dog
```

Fig - 6: Experimental result II.

In Figure 6, some samples of results are displayed. The first two images, cat1 and cat2, were predicted incorrectly, while the remaining images were predicted correctly.

4. Conclusion and Future Scope

After using the CNN model on the Kaggle Cats and Dogs dataset, the model was able to achieve a validation accuracy of 81.25 percent. On the 10th epoch, the model reached an accuracy of 81.25 percent. The accuracy of the results can be improved by training with a larger number of epochs. This model was trained with only 500 images of cats and dogs; hence, the result's accuracy can be improved by increasing the size of the dataset. This research provides immense knowledge about how images can be classified using CNN.

References

1. Md. Anwar Hossain & Md. Shahriar Alam Sajib, “Classification of Image using Convolutional Neural Network (CNN) “,Global Journals , Vol. 19, Issue 2, Version 1.0, year 2019.
2. Rahul Chauhan, Kamal Kumar Ghanshala, R.C Joshi, “Convolutional Neural Network (CNN) for Image Detection and Recognition”, International Conference on Secure Cyber Computing and Communication, R. C. Joshi, December 2018
3. P.Lakshmi Prasanna, D.Raghava Lavanya, T.Sasidhar, B.Sekhar Babu, “Image Classification Using Convolutional Neural Network”, International Journal of Emerging Trends in Engineering Research, Volume 8. No. 10, pp. 6816-6810, October 2020.
4. Muthukrishnan Ramprasath, M.Vijay Anand, Shanmugasundaram Hariharan, “Image Classification using Convolutional Neural Networks”, International Journal of Pure and Applied Mathematics, Volume 119 No. 17 2018, 1307-1319.
5. Atul Sharma,Gurbakash Phonsa, “ Image Classification Using CNN”, INTERNATIONAL CONFERENCE ON INNOVATIVE COMPUTING AND COMMUNICATION,year 2021.

AUTHORS PROFILE

Akshatha S.A earned her B. E degree in Electronics and Communication from SDIT Mangalore in 2014, and M.tech degree in Computer Networks Engineering from MITE Moodbidri. She is currently working as Assistant Professor in Department of Computer Science from Bearys Institute of Technology, Mangalore since 2021.Her main research work focuses on Big Data Analytics, Data Mining, IoT and Artificial Intelligence. She has 3 years of teaching experience.