WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Image Classification of Cats & Dogs

Prof. Sejal Thakkar[1], Dharmik Malaviya[2], Manav Madlani[3], Maitry Khammar[4], Karan Kapadia[5]

[1] *Assistant Professor. CE Department, Indus University, Ahmedabad, India.*
[2,3,4,5] *Student, B. Tech CE, Indus University, Ahmedabad, India.*

Abstract –

Over the past decade, computer vision has faced several accuracy limitations in various problems. However, the emergence of deep learning techniques has significantly improved the accuracy of these problems. One such major challenge was image classification, which involves predicting the class of an image, such as distinguishing between cats and dogs. This paper aims to enhance object detection accuracy by incorporating state-of-the-art techniques. To achieve this goal, the authors have developed a convolutional neural network specifically designed for image classification tasks.

I. INTRODUCTION

For numerous years, the classification problem of distinguishing between images of cats and dogs has been a matter of significant interest and attention. According to research, distinguishing between images of cats and dogs is a challenging task for automated systems, despite being relatively effortless for humans.

In this project, we intend to construct a convolutional neural network with the aim of achieving superior outcomes and enhancing performance in the task of differentiating between images of cats and dogs. Rather than utilizing the entire Kaggle dataset, which contains a total of 25,000 images, we have decided to work with a subset of these images for the purpose of my project. We plan to employ Keras for constructing the model, and Kaggle as the platform to implement all the code. The dataset will contain a total of 1800 images and will be used for the purpose of the project.

II. CONVOLUTIONAL NEURAL NETWORK

The term "convolutional neural network" indicates that the network leverages convolution, which is a specific type of linear operation in mathematics. Rather than using general matrix multiplication in all their layers, convolutional networks utilize convolution in at least one layer, making them distinct from traditional neural networks. In a convolutional neural network, there are an input layer and an output layer, as well as several hidden layers in between. These hidden layers typically contain a sequence of convolutional layers that carry out convolutions using a multiplication or dot product method.

Typically, a convolutional neural network employs the rectified linear unit (RELU) function as the activation function, which is subsequently followed by a series of additional convolutions such as pooling layers, normalization layers, and fully connected layers. These layers are often referred to as hidden layers because their inputs and outputs are masked by the activation function and final convolution, which allows the network to learn and extract features from the input images. The final convolutional layer of the network frequently utilizes backpropagation to refine the weighting of the output.

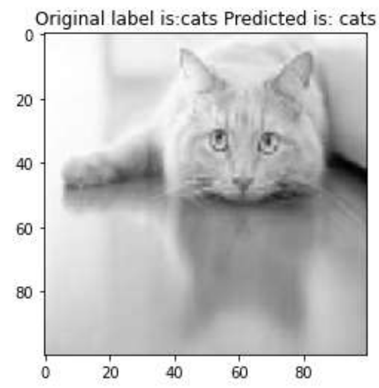


Fig. 1

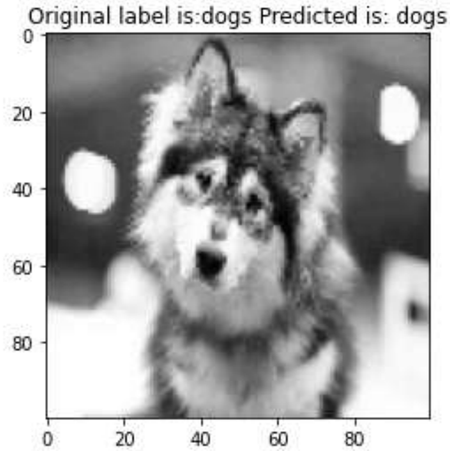


Fig. 2

III. DATASET AND DATA AUGMENTATION

The dataset in Keras is organized into folders for each class, with a training and testing set. Each set contains two folders, one for cat images and another for dog images. There are 4000 images of each cat and dog for training, and 1000 images of each for testing. However, the images in the dataset have varying shapes and sizes, which need to be standardized for training a convolutional neural network (CNN).

To address this, data augmentation techniques are applied using Keras' ImageDataGenerator module. The images are resized to 64 x 64 pixels during augmentation. Data augmentation is crucial for CNN training as it helps to increase the dataset size and improve model generalization. Techniques such as rescaling, shear range, and zoom range are utilized to augment the existing images and diversify the dataset. This allows the CNN to learn from a larger and more varied set of images, enhancing its ability to recognize patterns and generalize well to unseen data.

IV. MODEL ARCHITECHTURE

The purpose of the dropout layer is to randomly set a fraction of input units to 0 during training to prevent overfitting. Activation layers introduce non-linearity into the output of neurons, and Relu and sigmoid activation functions are used in this model. Batch normalization is used to improve the speed, performance, and stability of neural networks by normalizing the inputs of each layer.

The Conv2D layer applies a convolution matrix or mask to the input image, which can be used for blurring, sharpening, edge detection, and more. Fully connected layers follow the convolution layers, with only two layers in this model. The first layer is a global average pooling layer that minimizes overfitting by reducing the total number of parameters in the model. The second and final layer is a Dense layer with sigmoid activation. The Global Average Pooling 2D layer reduces the dimensionality of the feature maps by averaging all the values in each feature map to a single number. The Dense layer performs an operation that applies the activation function to the dot product of the input, weights matrix, and bias vector. This model has a total of 2,68,47,234 parameters, with 2,68,45,058 trainable parameters and 2176 non-trainable parameters

In summary, the given model is a Sequential model with Conv2D, batch normalization, activation, Conv2D, max pooling, dropout, and Dense layers. It uses activation functions like Relu and sigmoid, and batch normalization to improve performance. The model also has a Global Average Pooling 2D layer to reduce overfitting and a Dense layer with sigmoid activation as the final layer.

IV. MODEL COMPILATION AND TRAINING

During model compilation, there are several parameters to be considered such as the optimizer algorithm, loss function, and metrics. For this project, the Adam optimizer algorithm will be employed along with the binary cross entropy loss function, while accuracy will serve as the only metric. EarlyStopping and ModelCheckPointer callbacks will be utilized to prevent overfitting and to save the best state of the model during the training process.

The callbacks mentioned earlier are included in a list during the model training process. The Sequential models fit_generator () method will be utilized for training the model. During the training process, the model will be trained for 200 epochs while implementing EarlyStopping & ModelCheckPointer.

V. MODEL EVALUATION

The model was trained for a total of 18 epochs. During training, an EarlyStopping callback was employed with a patience parameter set to 5. As the training progressed, the training accuracy continued to improve. However, the validation accuracy started to decrease, indicating that the model might be overfitting. To prevent overfitting, the EarlyStopper check pointer was utilized. The results of the training process are summarized in a table showing the training accuracy, test accuracy, training loss, test loss, and number of epochs. must be placed after their associated figures, as shown in Fig. 1.

Table -1: Training Result Per Epoch

```
Epoch 45/50
18/18 [==============================] - 15s 817ms/step - loss: 0.3644 - accuracy: 0.8259 - val_loss: 0.8035 - val_accuracy: 0.64
29
Epoch 46/50
18/18 [==============================] - 14s 788ms/step - loss: 0.3595 - accuracy: 0.8492 - val_loss: 0.7530 - val_accuracy: 0.62
14
Epoch 47/50
18/18 [==============================] - 15s 838ms/step - loss: 0.3400 - accuracy: 0.8492 - val_loss: 0.7462 - val_accuracy: 0.60
71
Epoch 48/50
18/18 [==============================] - 15s 807ms/step - loss: 0.3265 - accuracy: 0.8636 - val_loss: 0.7067 - val_accuracy: 0.69
29
Epoch 49/50
18/18 [==============================] - 15s 856ms/step - loss: 0.3400 - accuracy: 0.8528 - val_loss: 0.7635 - val_accuracy: 0.65
71
Epoch 50/50
18/18 [==============================] - 15s 812ms/step - loss: 0.3140 - accuracy: 0.8528 - val_loss: 0.6791 - val_accuracy: 0.71
43
```

Fig. 3: Model Loss

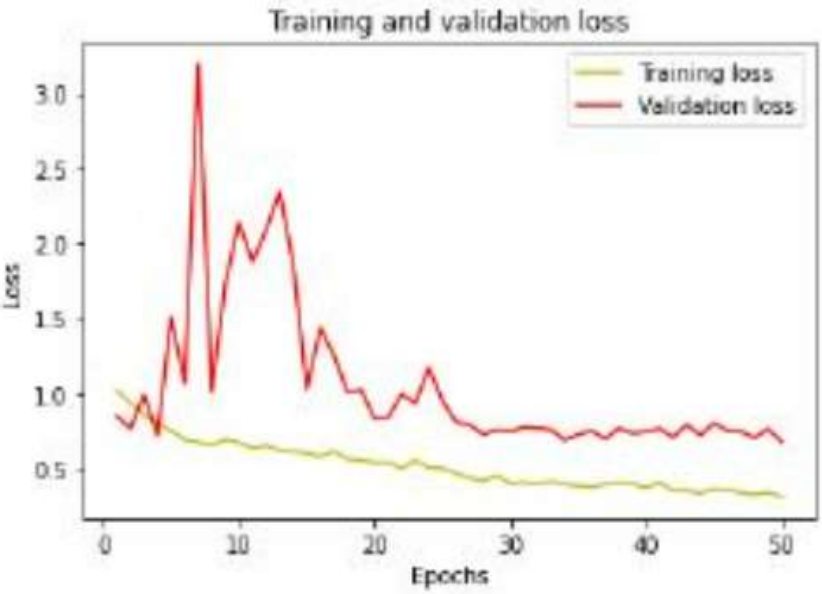
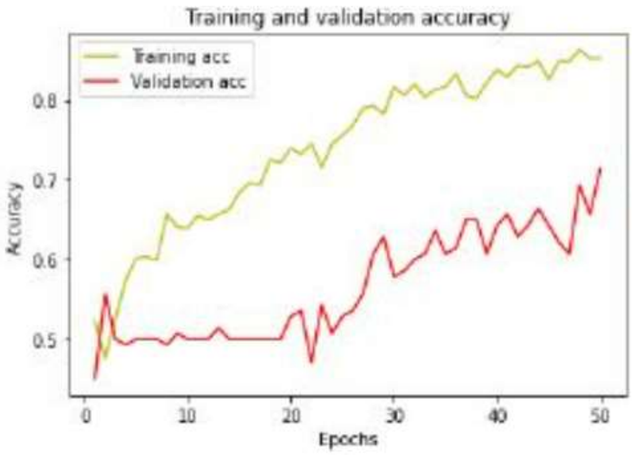


Fig. 4: Model Accuracy



CONCLUSION

In conclusion, the Image Classification model for Cat and Dog Images using CNN architecture has been successfully implemented. The Sequential model API has been utilized to create the model layer-by-layer, starting with a Conv2D layer, followed by batch normalization, activation, and max pooling layers. Dropout layers have also been included to prevent overfitting.

The model has been trained for a total of 18 epochs, and the EarlyStopping callback has been employed with a patience parameter of 5 to prevent overfitting. The final results show a training accuracy of around 85% and a test accuracy of around 71%, indicating that the model has achieved good performance on both the training and testing datasets.

The total number of parameters in the model is 2,68,47,234; out of which 2,68,45,058 are trainable parameters and 2176 are non-trainable parameters. The convolutional layers contribute to the majority of the trainable parameters.

Overall, the Image Classification model for Cat and Dog Images using CNN architecture has been successfully implemented with good performance on both the training and testing datasets.

REFERENCES

In 2008, Golle published a paper titled "Machine learning attacks against the Asirra CAPTCHA" in which he demonstrated the vulnerability of the Asirra system to automated attacks using machine learning techniques.

Elson, Douceur, Howell, and Saul (2007) introduced Asirra in their paper titled "Asirra: a CAPTCHA that exploits interest-aligned manual image categorization". The paper was presented at the Proc. of ACM CCS 2007.

Ramprasath, Anand, and Hariharan (2018) proposed the use of Convolutional Neural Networks for image classification in their research paper titled "Image Classification using Convolutional Neural Networks", published in the International Journal of Pure and Applied Mathematics.

In their research paper titled "Cats and dogs", Parkhi, Vedaldi, Zisserman, and Jawahar (2012) presented a large-scale dataset of cats and dogs' images for the purpose of image classification. The paper was presented at the Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference.

Zeiler and Fergus (2013) introduced a method for visualizing and understanding Convolutional Neural Networks in their paper titled "Visualizing and Understanding Convolutional Neural Networks", which was published on arXiv.

Liu, Liu, and Zhou (2016) proposed an image classification approach for dogs and cats in their paper titled "Image Classification for Dogs and Cats".