DISTINGUISH IMAGES OF DOG FROM A CAT

Project report submitted

As the requirement for the course of

'Computer Vision'

By

Ankit Yadav (170001007)	Vinayak Mohite (170004040)	Chaitanya Shah (170005007)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTIUTE OF TECHNOLOGY INDORE

(SEMESTER VII – ACADEMIC YEAR 2020-2021)

1. Problem Statement

Web services are often protected with a challenge that is supposed to be easy for people to solve, but difficult for computers. Such a challenge is often called a CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) or HIP (Human Interactive Proof). HIPs are used for many purposes, such as to reduce email and blog spam and prevent brute-force attacks on web site passwords.

Asirra (Animal Species Image Recognition for Restricting Access) is a HIP that works by asking users to identify photographs of cats and dogs. This task is difficult for computers, but studies have shown that people can accomplish it quickly and accurately.

By using modern computer vision approaches, the machine can solve the Asirra with accuracy greater than 80%. Therefore, Asirra is no longer considered safe from attack.

We will be implementing one of the approaches to crack the Asirra using the concepts in Machine Learning, Artificial Neural Network and Computer Vision.

2. Details of Dataset

The dataset is provided by *Petfinder.com*, the world's largest site devoted to finding homes for homeless pets. The images in the dataset are manually classified by the people.

The dataset comprises a total of 25000 .jpg images out of which 12500 are cats and 12500 are of dogs.

Link to the dataset:
https://www.kaggle.com/c/dogs-vs-cats/data

3. Workflow

3.1 Data Pre-processing

After observing the dataset, it was found that images are colour and have different sizes and shapes. Hence, we had to reshape all the images to a fixed size and transform them into grayscale having values in range [0, 1] (**normalization**).

A smaller image size means the model is faster to train but may affect accuracy. For the implementation purpose, we reshape images into 128 x 128 pixels.

The dataset has 25000 images. After loading them all into RAM, it took around >2GB of memory space. If the model is tuned in the future, then we again have to load all the images into memory and this takes a lot of time. To avoid this, after processing the dataset (i.e., after reshaping and train-test splitting) we stored the processed dataset in storage.

We applied some image transformation operations (like flipping image horizontally, adding some shear, or zooming a little) on our dataset images to increase the diversity of data available (**augmentation**). Therefore, later we had 30000 images for training and 7500 for testing our image classification model.

3.2 Developing the model

The architecture requires stacking convolutional layers with small 3×3 filters followed by a max-pooling layer. Together, these layers form a block, and these blocks can be repeated where the number of filters in each block is increased with the depth of the network such as 32, 64, 128, 256 for the first four blocks of the model. There is no Padding done on the convolutional layers, so the height and width shapes of the output feature maps are less than the inputs.

Each layer will use the ReLU activation function except the last layer. The last layer uses SoftMax activation function for binary classification.

We compared the performances of the three models. The first model contains two blocks. In the second model, we added one more block. The third model consisted of a total of 4 blocks.
Here one block = convolutional layer + max-pooling layer.

Model 2 and 3 may overfit the data because of large number of neurons and weights. To avoid this after each block we have used the dropout technique to randomly ignore selected neurons. This results in avoiding overfitting and also reduces the training time.

We have used Adam Optimizer.

Why Adam Optimizer?

- Quite computationally efficient.
- Requires little memory space.
- Works well with large data sets and large parameters.

Learning rate is 0.001.

Loss function used is sparse categorical cross-entropy.

3.2.1 Two block model

2 Convolution layer + 2 max-pooling layer

3.2.2 Three block model with dropout

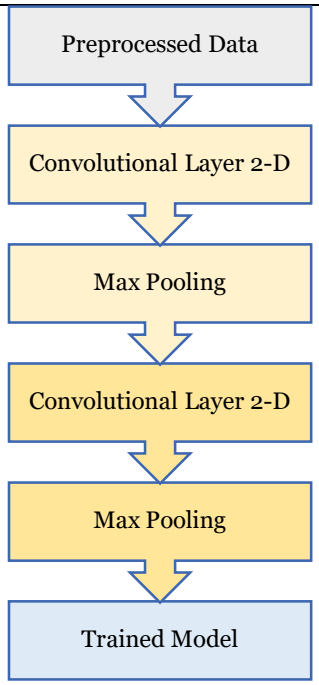
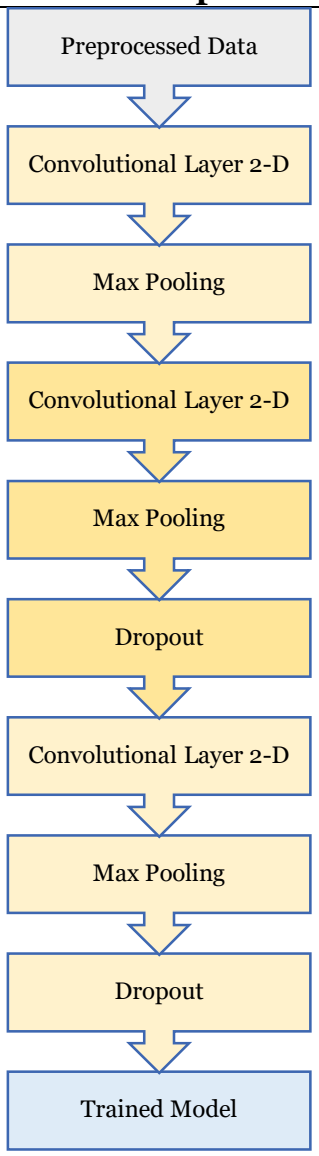
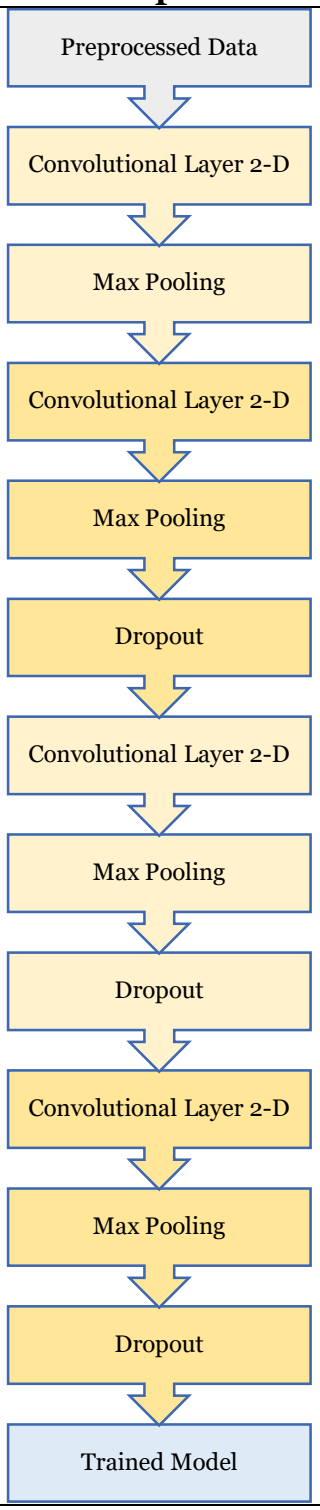
3 Convolution layers + 3 max-pooling layers + Dropout

Here we have added dropout with value 0.4 after every block. This means that the probability that the selected neuron will be dropped will be 25%.

3.2.3 Four block model

4 Convolution layers + 4 max-pooling layers + Dropout

Here we have added dropout with value 0.25 after every block. This means that the probability that the selected neuron will be dropped will be 25%.

Two Block Model	Three Block Model with Dropout	Four Block Model with Dropout
		

4. Results and Analysis

Model No.	Model	Accuracy
1	Two block model	74.64 %
2	Three block model with dropout	81.33 %
3	Four block model with dropout	90.34 %

Model 1 doesn't perform well on testing data. It has a high variance and hence underfits the data.
Model 2 performs better than model 1.
Model 3 has very low variance and bias and hence performs the best.

5. Conclusion

We explored how to develop a convolutional neural network to classify photos of dogs and cats.

Specifically, we learned:

- How to load and prepare photos of dogs and cats for modelling.
- How to develop a convolutional neural network for photo classification from scratch and improve model performance.
- How to tune hyper-parameters
- Data Augmentation effect on model performance

6. References

i. https://www.kaggle.com/c/dogs-vs-cats
ii. https://www.irjet.net/archives/V6/i12/IRJET-V6I1271.pdf
iii. https://en.wikipedia.org/wiki/Convolutional_neural_network