Major Project Report - PETFINDER.MY –PAWPULARITY

Shyam Sundar Murali Krishnan

December 13 2021

Chapter 1

Introduction

This project is an application project. Everyday there are lots of stray animals on streets which suffer or get killed humanely in shelters around the world. We could think that if a nice picture of these animals are available it would develop more interest to the people and they can adopt it faster. But the question is what is a good picture. What are the traits that define a good picture and how we can identify it.

To answer these questions we would need effective machine learning models, training and testing the models on different sets of images for predicting and analyzing the research questions mentioned above. PetFinder.my is a animal welfare platform that currently uses a basic cuteness meter to rank the pet photos. While this tool seems to be useful there are lots of improvements that could be made and this functionality is still in experimental stage. This project is for that welfare platform.

Since in this project we need to find out is the picture nice enough to gain interest it looks more like a image classification problem. Basically in image classification the images are segmented into different categories based on features.

The features will be like pixel intensity, change in pixel values, color composition, background color and many more. For implementing this kind of problem I am planning to use the Convolution Neural network model which is a part of Deep Learning model. The reason for using this model is because it assigns importance to various aspects of the image and helps in differentiating one another.

Chapter 2

Methodology

In this chapter the methodology used for this project is explained in detail. The steps are pre-image processing, Convolution layer, Max-pooling layer, Flattening layer. The Convolution neural network is executed in python and tensor-flow package was used to create the network for training and testing.

2.1 Image Pre-Processing

Since the images will be portrait and landscapes it will have different sizes. So in order make every image as a same size I resized all the images into 150 X 150 X 3. The images used as an input are the RGB images.

2.2 Convolution Layer

The convolution layer extracts features from input image. It preserves relationship between pixels by learning image features using small squares of data.This layer does mathematical operation that takes two inputs such as image matrix and a filter.

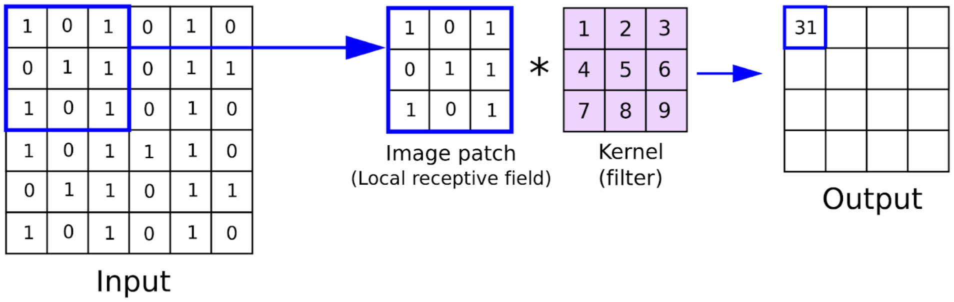


Figure 2.1: Convolution Layer

2.3 Max-Pooling Layer

This layer reduces dimensions of the output from the convolution layer. It reduces number of parameter to learn. It reduces amount of computation.

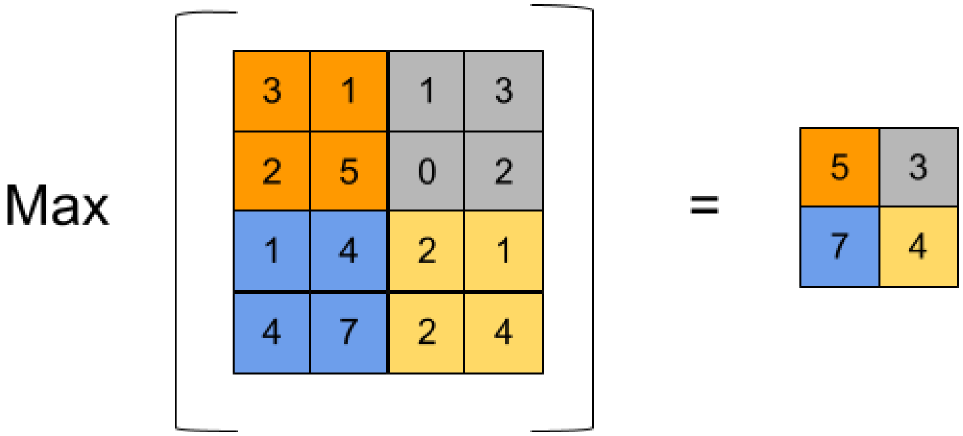


Figure 2.2: Max-Pooling Layer

2.4 Flattening Layer

This layer converts data to 1- dimensional array. It inputs for next coming dense layer or fully connected layer.

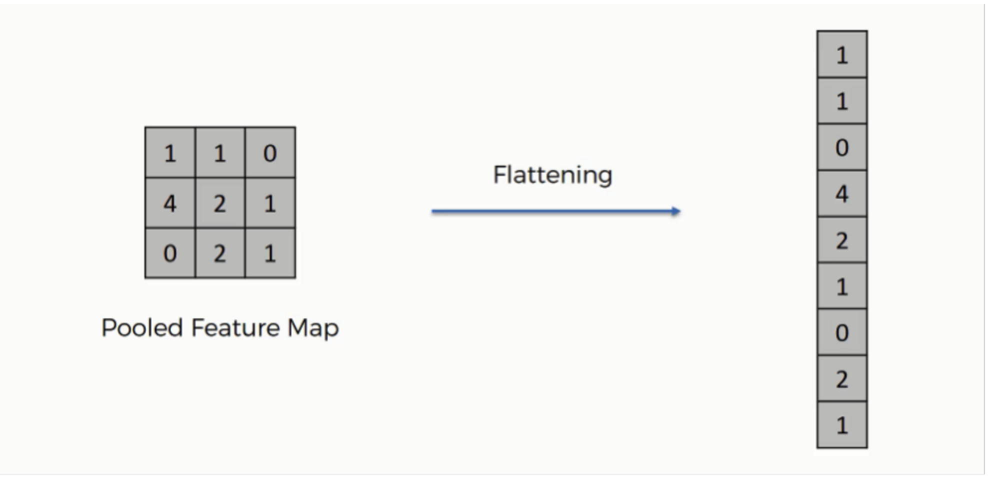


Figure 2.3: Flattening Layer

Chapter 3

Experiment

In this chapter I will explain about the data prepared for the project followed by the experimental setup followed by observation and discussion about the results obtained.

3.1 Data Preparation

The data for this project is given from PathFinder.my which consists of 9912 images of pets pictures posted on their platform. They also give a metadata for each image which consists of the label which is called as the Pawpularity Score. The score ranges from 0-100. In this experiment I converted into binary classification by considering 1 if pawpularity score is greater than or equal to 50 and 0 if the score is less than 50. 70% of data is used for training and 30% of the data is used for validation.

The testing images are collected from different sources like the kaggle, google pictures and it is around 3000 images of pet pictures.

3.2 Experimental Setup

The activation function used in the convolution layers are the ReLU activation function. The activation function for the output layer is the sigmoid activation function since the output is binary. The training data is 70% of the total dataset and the validation data is 30% of the total dataset.

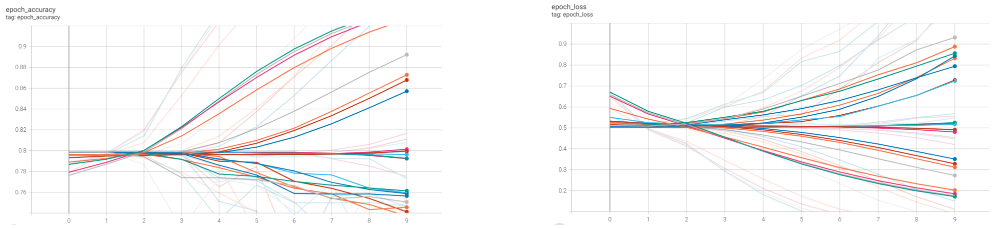
The filter size used for this experiment is 3 X 3 and the pooling size used is 2 X 2. The type of pooling used in this experiment is Max - pooling layer since it has been used widely in most of the image recognition applications for better prediction.

The number of convolution layers are chosen to 3 and the number of nodes in the layers are chosen to be 128. There are no dense layer taken into consideration for this experiment. The output layer has one node which output if the picture is good enough (1) or not (0). The number of layers, nodes and dense layers are chosen by conducting experiments on different models and choosing the best one which has not only good prediction accuracy but also good evaluation accuracy.

3.3 Results and Discussions

The experiment was conducted for around 15 models with different number of convolution layer, number of nodes in the layer and number of dense layers in the network. There were many models which gave prediction accuracy of about 98% but gave a very low evaluation accuracy of around 50%. This might be because of overfitting problem where the in samples predicted well whereas the out samples did not give a very good prediction. This means that the models are trying to memorize all the samples it has gone through and doesn't really learn from it for efficient prediction.

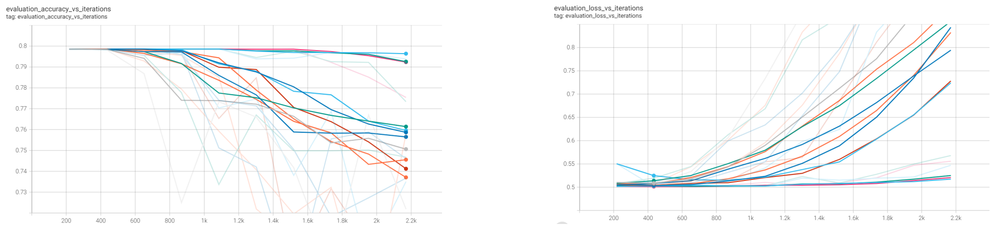
So, considering these pitfalls in different models in mind a model was selected to be the best model compared to others which had a prediction accuracy of 82% and the evaluation accuracy of around 80% and this model had 3 convolution layers, 128 nodes for each layer and no dense layer. This model when used on the testing data gave an accuracy around 79 %.



(a) Epoch Accuracy.

(b) Epoch Loss.

Figure 3.1: Epoch Accuracy and Loss



(a) Evaluation Accuracy.

(b) Evaluation Loss.

Figure 3.2: Evaluation Accuracy and Loss