Guru Nanak Dev Engineering College

Ludhiana

Department Of Information Technology

Project Synopsis

For Minor Project

Batch 2018-2022

Session Jan-June 2021

Subject Of Project : IMAGE CLASSIFICATION

Submitted By :

Lovish Rajpal(1805522)

Gursimran Singh(1805512)

Contents

Project Synopsis

March 17, 2021

1 Introduction

Image classification refers to a process in computer vision that can classify an image according to its visual content. it is the labelling of images into one of a number of predefined classes. There are potentially n number of classes in which a given image can be classified. Manually checking and classifying images could be a tedious task especially when they are massive in number and therefore it will be very useful if we could automate this entire process using computer vision. The advancements in the field of autonomous driving also serve as a great example of the use of image classification in the real-world. The applications include automated image organization, stock photography and video websites, visual search for improved product discoverability, large visual databases, image and face recognition on social networks, and many more; which is why, we need classifiers to achieve maximum possible accuracy.

2 Objective

The image classifier performs the role of a discriminant – discriminates one class against others Discriminant value highest for one class, lower for other classes (multiclass). Discriminant value positive for one class, negative for another class (two class).

The goal of this post is to showconvnet (CNN — Convolutional Neural Network) works. I will be using classical cat/dog classification example. An analogy can be described with the way how humans think. Each of us knows how airplane looks, but most likely when thinking about airplane we are not thinking about every little bit of airplane structure. In a similar way, convnet learns to recognize higher level elements in the image and this helps to classify new images when they look similar to the ones used for the training. IN THIS PROJECT WE WILL BE DOING CATS AND DOGS IMAGE CLASSSIFICATIONS USING trensorflow from kaggle machine learning. The Dogs vs. Cats dataset is a standard computer vision dataset that involves classifying photos as either containing a dog or cat. Although the problem sounds simple, it was only effectively addressed in the last few years using deep learning convolutional neural networks. While the dataset is effectively solved, it can be used as the basis for learning and practicing how to develop, evaluate, and use convolutional deep learning neural networks for image classification from scratch.

3 Feasibility

Image Classification is also done by Multilayer Perceptron on centered images of dogs and cats and fails if it is slightly aside from the centre like if it is on top right corner or on left corner its prediction will fail. The performance was pretty good 98.3 percent accuracy was achieved on test data. But there was a problem with that approach. All images that are centered, then the MLP approach fails miserably. We want

the network to be Translation-Invariant.. The classifier predicts correctly for the centered image but fails in any other cases. To make it work for other images that are not centered, either we have to train separate MLPs for different locations or we have to make sure that we have all these variations in the training set as well, which I would say is difficult, if not impossible. These problems are solved to a great extent by using Convolutional Neural Networks which we arguing to use in image classification of dogs and cats.

4 Methodology(Planning to work)

let's take a moment to understand how an image classification model is typically designed. We can divide this process broadly into 4 stages. Each stage requires a certain amount of time to execute:

Loading and pre-processing Data – 30% time Defining Model architecture – 10 % time Training the model – 50% time Estimation of performance – 10% time Let me explain each of the above steps in a bit more detail. This section is crucial because not every model is built in the first go. You will need to go back after each iteration, fine-tune your steps, and run it again. Having a solid understanding of the underlying concepts will go a long way in accelerating the entire process.

Stage 1 Loading and pre-processing the data:

Data is gold as far as deep learning models are concerned. Your image classification model has a far better chance of performing well if you have a good amount of images in the training set. Also, the shape of the data varies according to the architecture/framework that we use.

Hence, the critical data pre-processing step (the eternally important step in any project). I highly recommend going through the 'Basics of Image Processing in Python' to understand more about how pre-processing works with image data.

But we are not quite there yet. In order to see how our model performs on unseen data (and before exposing it to the test set), we need to create a validation set. This is done by partitioning the training set data.

In short, we train the model on the training data and validate it on the validation data. Once we are satisfied with the model's performance on the validation set, we can use it for making predictions on the test data.

Time required for this step: We require around 2-3 minutes for this task.

Stage 2 Defining the model's architecture:

This is another crucial step in our deep learning model building process. We have to define how our model will look and that requires answering questions like:

How many convolutional layers do we want? What should be the activation function for each layer? How many hidden units should each layer have? And many more. These are essentially the hyperparameters of the model which play a MASSIVE part in deciding how good the predictions will be.

How do we decide these values? Excellent question! A good idea is to pick these values based on existing research/studies. Another idea is to keep experimenting with the values until you find the best match but this can be quite a time consuming process.

Time required for this step: It should take around 1 minute to define the architecture of the model.

Stage 3 Training the model:

For training the model, we require:

Training images and their corresponding true labels Validation images and their corresponding true

labels (we use these labels only to validate the model and not during the training phase) We also define the number of epochs in this step. For starters, we will run the model for 10 epochs (you can change the number of epochs later).

Time required for this step: Since training requires the model to learn structures, we need around 5 minutes to go through this step.

And now time to make predictions!

Stage 4 Estimating the model's performance:

Finally, we load the test data (images) and go through the pre-processing step here as well. We then predict the classes for these images using the trained model.

Time required for this step: 1 minute.

5 Facilities required for proposed work

Hardware requirements are : an Operating System, RAM, HDD
Software requirements are : All you need is Python compiling IDE, The one which is User friendly and Easy to use is Pycharm. You should have installed some of the Modules/Packages like Keras,TensorFlow.
These functions are particularly required in Image Classification:-
from keras.models import Sequential
from keras.layers import Convolution2D,MaxPool2D,Flatten,Dense
from keras.preprocessing.image import ImageDataGenerator

6 References

https://towardsdatascience.com/cat-or-dog-image-classification-with-convolutional-neural-network-d421a9363c7a
https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-to-classify-photos-of-dogs-and-cats/
https://learnopencv.com/image-classification-using-convolutional-neural-networks-in-keras/