Proposal towards

Udacity Machine Learning Engineer Nanodegree capstone project

By Siddharth Shorya <shoryasiddharth90@gmail.com>

23th July 2018

1. Domain Background

Deep learning is a part of a broader family of machine learning methods based on learning data representation, as opposed to task- specific algorithm.[1] Deep learning attempts to model high-level abstractions about data using networks of graphs i.e. modelling high-level abstractions about data similar to AI-the idea that knowledge can be represented and acted upon intelligently.

Deep learning architectures such as **deep neural networks**, **convolutional neural networks** and **deep models for sequence learning** have been applied to fields including **computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design and board game programs**, where they have produced results comparable to and in some cases superior to human experts.[1]

This was in 1980-1990 when neural networks were invented but then they didn't work because neural nets need a lot of data. In that time there was a very less data and slow computers. But now from last few years since we have a lot of data and cheap GPU's NN are doing wonders 2009 came up with speech recognition, 2012 came up with computer vision and 2014 with machine translation, most of them are not mentioned.

My proposal is based on using CNNs. CNNs have been extremely successful in computer vision applications, such as face recognition, object detection, powering vision in robotics, and self-driving cars.[2] Wildlife images captured are given for recognition of species which is very tough task as pictures are captured in different light, climate and position. Here is the place where CNN can be used to make the task easy. Monitoring wild animals is essential as it provides researchers evidences to inform conservation and management decisions to maintain diverse, balanced and sustainable ecosystems in the face of those changes.[3] This research is important as this will let us know exactly what it is, as many of the animals look similar and no doubts anyone can make mistakes in identification.

1.1 Motivation

My interest towards predictive analysis brought me to ML and now I am drawn to deep learning because I find it as an exciting branch of ML that uses data, to teach computers how to do thing only humans were capable of before.

2. Problem Statement

For my MLND Capstone project, I propose to explore the domain of deep learning and apply it to the problem of identifying an animal.

Wildlife images captured in a field represent a challenging task while classifying animals since they appear with a different pose, cluttered background, different light and climate conditions, different viewpoints, and occlusions. Additionally, animals of different classes look similar. Also, processing such a large volume of images and videos captured from camera traps manually is extremely expensive, time-consuming and also monotonous. All these challenges necessitate an efficient algorithm for classification.[4]
In this project, I will train a neural network to predict the animal for the ease and accurate classification.

3. Datasets and inputs

I have two types of files CSV and Images. The train data consists of 13,000 images and the test data consists of 6,000 images of 30 different species of animals. The image ID and the corresponding animal name are stored in .csv format, while the image files are sorted into separate train and test image folders. Data in the .csv file is in the following format:

Variable	Description
Image_id	Image name
Animal	Name of the Animal

Following are the 30 different species of animals in the dataset:

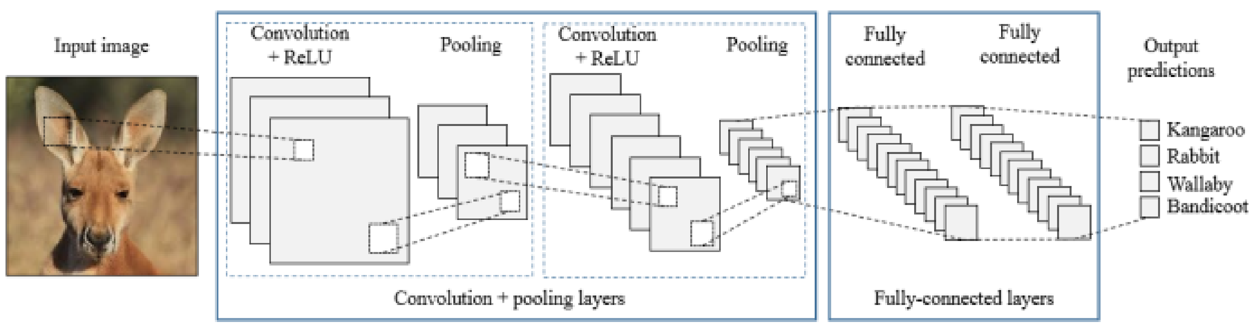
- antelope
- bat
- beaver
- bobcat
- buffalo
- chihuahua
- chimpanzee
- collie
- dalmatian
- german + shepherd
- grizzly + bear
- hippopotamus
- horse
- killer + whale
- mole
- moose
- mouse
- otter

- ox
- Persian + cat
- raccoon
- rat
- rhinoceros
- seal
- Siamese + cat
- spider + monkey
- squirrel
- walrus
- weasel
- wolf

The input will be the images 224 x 224 (will convert high resolution to lower resolution) and my program will be giving the possible probabilities of all the species. I'll train as well as show the test results.

4. Solution statement

The solution to the above-mentioned problems is Convolutional Neural Networks (CNN). Convolutional NN have many hidden layers between the input and output. The settlement of these layers and their number matter a lot for accuracy.



In this problem solving I will use the following CNN architecture-

- Three 2-D convolutional layers with ReLU activations and MaxPooling
- Followed by two fully-connected layers: one with eLU nonlinear activation plus Dropout for reducing overfitting.
- All convolutional layers with small filter size of 3 × 3, while all max-pooling layers with window size of 2×2 pixels.

According to me this architecture would give me wonderful results.

5. Benchmark model

While researching this project, I came across approaches for image segmentation. Image segmentation is one of the mostly used methods to classify the pixels of an image correctly in a decision-oriented application. It divides an image into a number of discrete regions such that the pixels have high similarity in each region and high contrast between regions[5]. This is what I'll do with CNN but most of the approaches I came across were using K-means clustering algorithm.

Undoubtably among many techniques K-means is one the most efficient algorithm to use and thus I propose to use it as a benchmark.

6. Evaluation metrics

- **Accuracy**: When an image will there for predicting the animal then it may predict right or wrong, so how much percentage of predictions is right will be the accuracy.

7. Project design

For this problem, I'll start with downloading the dataset. The datasets that contain 19,000 images of 30 different animal species. Then I will pre-process the images. The images will be normalized and labels will be one-hot encoded.

Once this above work will be done things will be ready to apply convolutions.

I then plan to make functions for all specific layers as this will make thing less complicated and easy to implement. I will use the following CNN architecture-

- Three 2-D convolutional layers with ReLU activations and MaxPooling
- Followed by two fully-connected layers: one with eLU nonlinear activation plus Dropout for reducing overfitting.
- All convolutional layers with small filter size of 3 × 3, while all max-pooling layers with window size of 2×2 pixels.

I just plan to use eLU as the non-linear function but can use ReLU too, depends on which gives me better results.

In the end I'll look for predictions and use the testing data to find the accuracy of my model.

References:

1. https://en.wikipedia.org/wiki/Deep_learning
2. https://www.hindawi.com/journals/cin/2018/7068349/
3. https://ieeexplore.ieee.org/document/8259762/
4. https://www.researchgate.net/publication/319392368_Animal_Recognition_and_Identification_with_Deep_Convolutional_Neural_Networks_for_Automated_Wildlife_Monitoring
5. https://www.sciencedirect.com/science/article/pii/S1877050915014143