

Image recognition using Machine learning

Abhinav N Patil

B tech student, Dept of S&T, Vishwakarma University

Abstract

Image recognition is necessary side of image processing for machine learning without involving any human support at any step. In this paper we study how image classification is done using imagery backend. Couple of thousands of images of each, cats and dogs are taken and then distributed them into category of test dataset and training dataset for our learning model. The results are obtained using custom neural network with the architecture of Convolution Neural Networks and Keras API.

Keywords: CNN, Keras, Machine-learning, Deep-learning, Keras, Image-recognition

1. Introduction

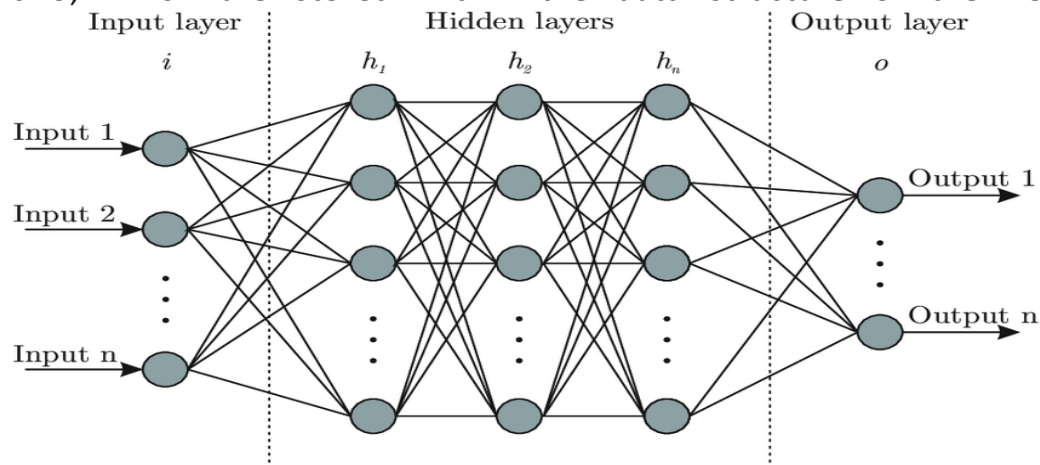
Image classification came into existence for decreasing the gap between the computer vision and human vision by training the computer with the data. **Artificial Intelligence** has for decades been a field of research in which both scientists and engineers have been making intense efforts to unravel the mystery of getting machines and computers to perceive and understand our world well enough to act properly and serve humanity. One of the most important aspect of this research work is getting computers to understand visual information (**images and videos**) generated everyday around us. This field of getting computers to perceive and understand visual information is known as computer vision. During the rise of artificial intelligence research in the **1950s** to the **1980s**, computers were manually given instructions on how to recognize images, objects in images and what features to look out for. This method are traditional algorithms and were called **Expert Systems**, as they require that humans take the pain of identifying features for each unique scene of object that has to be recognize and representing these features in mathematical models that the computer can understand. That involves a whole lot of tedious work because there are hundreds and thousands of various ways an object can be represented

and there are thousands (or even millions) of different scenes and objects that uniquely exist, and therefore finding the optimized and accurate mathematical models to represent all the possible features of each objects or scene, and for all possible objects or scene is more of work that will last forever. In the **1990s**, the concept of **Machine Learning** was introduced and it ushered in an era in which instead of telling computers what to look out for in recognizing scenes and objects in images and videos, we can instead design algorithms that will make computers to learn how to recognize scenes and objects in images by itself, just like a child learns to understand his/her environment by exploring. Machine learning opened the way for computers to learn to recognize almost any scene or object we want them too.

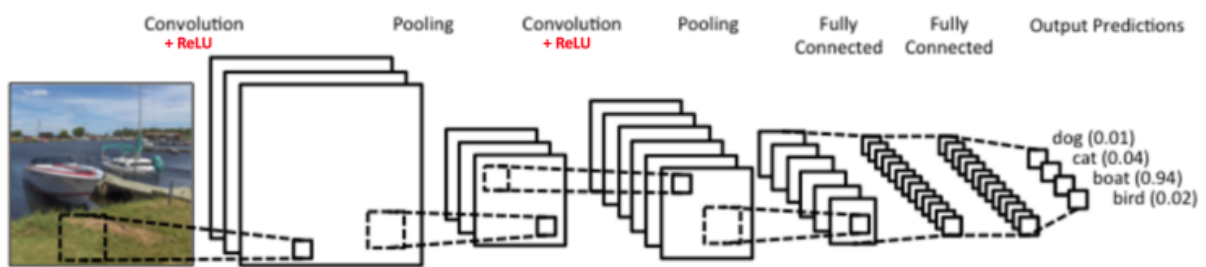
The basic artificial neural network is outlined in Section-II. Section-III describes about Convolutional Neural Network. The implementation and results are discussed in Section-IV. We conclude in section-V and finally the references are given at the end.

2. Artificial Neural Network

A neural network is a combination of hardware bonded or separated by the software system which operates on the small part in the human brain called as neuron. A multi layered neural network can be proposed as an alternative of the above case. The training images samples should be more than nine times the number of parameters essential for tuning the classical classification under very good resolution. The data structures and functionality of neural nets are designed to simulate associative memory. Neural nets learn by processing examples, each of which contains a known "input" and "result," forming probability-weighted associations between the two, which are stored within the data structure of the net itself.

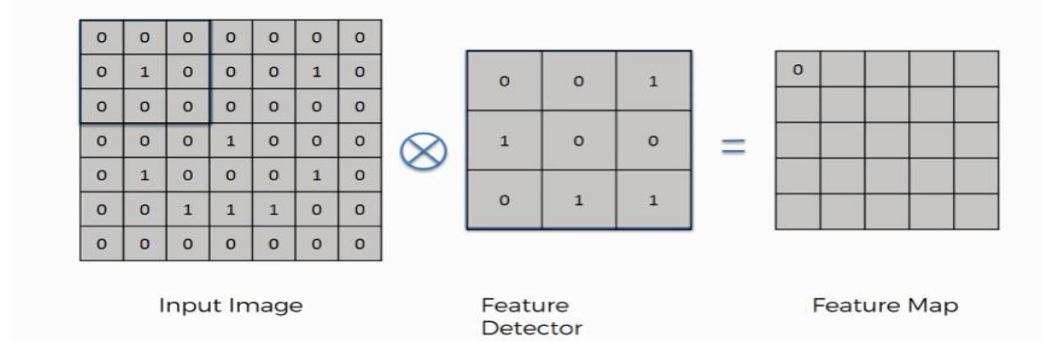


We train the model by providing inputs which then pass through hidden cycles which custom grid images and extract data from each specific section and conveying the network about its output. Neural networks are expressed in terms of number of layers involved for producing the inputs and outputs and the depth of the neural network. The Convolutional neural network is the most famous for implementing genetic algorithms for hidden layers which involve pooling and padding of data to make them ready through test dataset to be inserted into training model.



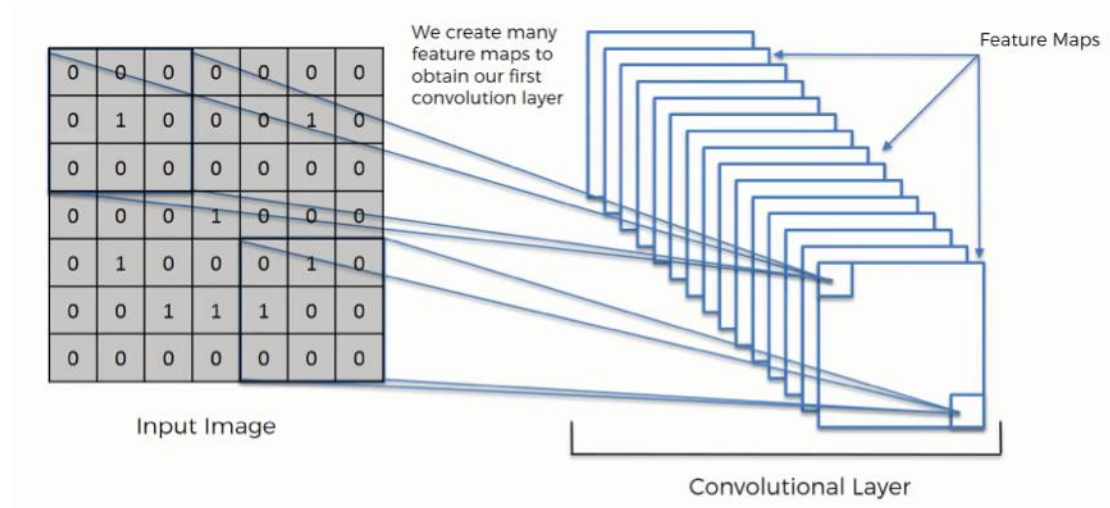
3. Convolutional Neural Network

In deep learning, a convolutional neural network (CNN, or ConvNet) is class of deep neural networks, most commonly applied to analysing visual imagery. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics. They have applications in image and video recognition, recommender systems, image classification, medical image analysis, natural language processing, and financial time series. The network places a 3x3 cell matrix over the provided image and then converts the data into a feature map with 1s and 0s, this operation is repeated for the whole image and feature maps are created with each layer having better feature detector applied.



Through training, the network determines what features it finds important in order for it to be able to scan images and categorize them more accurately.

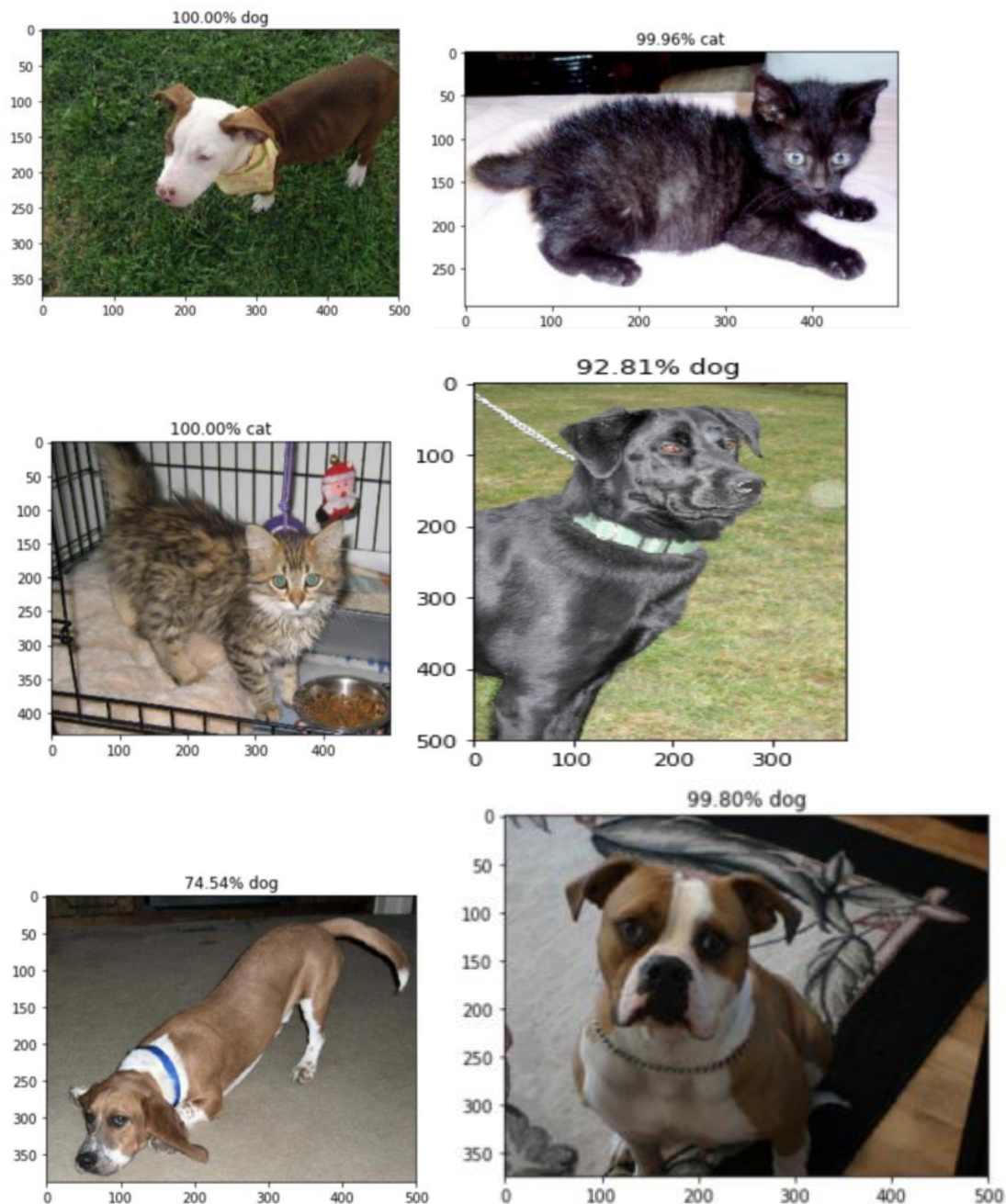
Based on that, it develops its feature detectors. In many cases, the features considered by the network will be unnoticeable to the human eye, which is exactly why convolutional neural networks are so amazingly useful. With enough training, they can go light years ahead of us in terms of image processing.



4. Implementation, Results and Discussion

We selected total of nearly 24,000 images of cats and dogs and also having rotations and scaling of many images as each scaled image is a new image for the model as input. For training the model we kept around 90% of dataset and rest for testing it extensively. 10 epochs (training cycles) were provided for images and with every epoch, the accuracy was expected to be increased which uses the power of TensorFlow backend dependent of discrete GPU. In the first layer we obtained the output volume size of 55x55x96. The model is trained on the GPU named GTX 1050ti which provides 4gb of memory. So, the first layer is sent to the GPU after converting it to 55x55x48. The 2nd, 4th, 6th, 8th and 10th layers are the bits which already dwell on related feature maps of each of its previous layer. The inputs are then flattened, and applied dense filter by providing matrix of 128x128 which provided 77.8% accuracy, although we noticed that accuracy can be improved by adding more layers/neurons to the model. The training cycles were increased to 20 from 10 and lowering the learning rate from 0.001 to 0.0001 and also added high level filter (256 from 128).

After checking the performance, our accuracy bumped up to 88% after having 20 cycles and again accuracy increased to 93% by increasing the filter size and decreasing the learning rate. These were training matrices and we were yet to apply them on our test dataset, we increased our cycles a bit more while they provided accuracy of up to 97.3% while taking tremendous amount of time and GPU power. The results as you can see below,



As we noticed, results do fluctuate a bit but according to the average, the accuracy was well around 90-95% percent with a layer filter of 256, thus

more powerful hardware could definitely achieve even higher results and with much extended dataset for more categories than just two for training.

5. Conclusion

The testing of random images came out to be successful. The image dataset was pulled from google repository directly. The convolutional neural network is used in-hand with Keras for classification purpose. From the experiments we observe that the images are classified correctly even if the same images were scaled in different sizes or trimmed or rotated to get entirely new image for the input showing the effectiveness of deep-learning algorithm.

References

1. Thomas, Elsken; Jan Hendrik, Metzen; Frank, Hutter (2017-11-13). "Simple and Efficient Architecture Search for Convolutional Neural Networks"
2. Elsken, Thomas; Metzen, Jan Hendrik; Hutter, Frank (2018-04-24). "Efficient Multi-objective Neural Architecture Search via Lamarckian Evolution".
3. Springenberg, Jost Tobias; Dosovitskiy, Alexey; Brox, Thomas; Riedmiller, Martin (2014-12-21). "Striving for Simplicity: The All Convolutional Net".
4. Romanuke, Vadim (2017). "Appropriate number and allocation of ReLUs in convolutional neural networks". *Research Bulletin of NTUU "Kyiv Polytechnic Institute"*. **1**: 69–78.
5. Bengio, Yoshua; Lamblin, Pascal; Popovici, Dan; Larochelle, Hugo (2007). "Greedy Layer-Wise Training of Deep Networks"(PDF). *Advances in Neural Information Processing Systems*: 153–160.
6. "Using TPUs | TensorFlow". *TensorFlow*. Retrieved 2018-11-14.
7. "Keras Documentation". *keras.io*. Retrieved 2016-09-18.
8. TensorFlow (March 30, 2018). "Introducing TensorFlow.js: Machine Learning in JavaScript". *Medium*. Retrieved May 24, 2019.