

Image Classification: A Survey

Ankita Singh¹, Pawan Singh²

^{1,2}Department of Computer Science and Engineering, Amity University Uttar Pradesh Lucknow Campus, India
¹ankitasingh8019@gmail.com, ²pawansingh51279@gmail.com

How to cite this paper: A. Singh, P. Singh (2020) Image Classification: A Survey. *Journal of Informatics Electrical and Electronics Engineering*, Vol. 01, Iss. 02, S. No. 2, pp. 1-9, 2020.

<http://doi.org/10.54060/JIEEE/001.02.002>

Received: 27/10/2020
Accepted: 11/11/2020
Published: 19/11/2020

Copyright © 2020 The Author(s).
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The Classification of images is a paramount topic in artificial vision systems which have drawn a notable amount of interest over the past years. This field aims to classify an image, which is an input, based on its visual content. Currently, most people relied on hand-crafted features to describe an image in a particular way. Then, using classifiers that are learnable, such as random forest, and decision tree was applied to the extract features to come to a final decision. The problem arises when large numbers of photos are concerned. It becomes a too difficult problem to find features from them. This is one of the reasons that the deep neural network model has been introduced. Owing to the existence of Deep learning, it can become feasible to represent the hierarchical nature of features using a various number of layers and corresponding weight with them. The existing image classification methods have been gradually applied in real-world problems, but then there are various problems in its application processes, such as unsatisfactory effect and extremely low classification accuracy or then and weak adaptive ability. Models using deep learning concepts have robust learning ability, which combines the feature extraction and the process of classification into a whole which then completes an image classification task, which can improve the image classification accuracy effectively. Convolutional Neural Networks are a powerful deep neural network technique. These networks preserve the spatial structure of a problem and were built for object recognition tasks such as classifying an image into respective classes. Neural networks are much known because people are getting a state-of-the-art outcome on complex computer vision and natural language processing tasks. Convolutional neural networks have been extensively used.

Keywords

Machine Learning, Deep Learning, Deep Neural Networks, Convolutional Neural Network, CIFAR-10 Dataset

1. Introduction

1.1. Machine learning

Machine learning is an operation that requires teaching any computer system how to forge accurate predictions when any kind of data has been fed into the computer system [1]. These predictions could be the answer to whether a piece of vegeta-

ble in a photo is broccoli or a beetroot, analyzing Twitter comments as negative, positive, or neutral, predicting stock prices of the market, whether an email is a spam or not, or recognizing a speech accurately to generate some captions for a video. The important difference between traditional computer software and a machine learning model is that humans have not written any code that tells the computer system how to tell the difference between broccoli and Beetroot. Instead, the machine-learning model has been taught how to accurately differentiate between the Vegetables by training the model on a large amount of data, for instance, a great number of photos with vegetables.

1.2. Deep learning

Deep learning is a sub-division of machine learning. It is a branch that is found on learning and ameliorating on its very own by inspecting certain computer algorithms. While machine learning utilizes easier concepts, here deep learning performs with various neural networks that are constructed to emulate how a human learns and thinks. Until now, these neural networks were just restricted by the computing power and consequently were confined in complexity [11]. Advancements in big data analytics have allowed greater, sophisticated neural networks, allowing the computers to think, learn, and react to byzantine situations very faster than humans. Deep learning has assisted in the classification of images, translation of language, etc.

2. Types of Machine learning

2.1. Supervised Learning

In machine learning, such type of learning is the type where we can see that the learning is assisted by someone, for example, a professor. So, we have a dataset that acts as the professor, and the role of a dataset i.e. the professor is to train the model or the machine. After the model gets trained, the model can then process making a prediction, the decision when the new data is fed to it.

2.2. Unsupervised Learning

A model learns through the observations and finds structure in those data. When the model is given the dataset, it can automatically find a certain pattern and relationship in that dataset by constructing some clusters in it. What the model cannot do is that it can't add a label to clusters, for example, it could not say that this is a group of vegetables or fruits, however, it can discriminate all the vegetables from the fruits. For instance, we present some images of cars, trucks, and planes to a model, so what it will do, depending on some patterns and relationships, will create some cluster and divide that particular data set into those clusters. Later on, if fresh data is put into the model as input, it puts that into one of the already built clusters.

2.3. Reinforcement learning

It is a potential of an agent, to inter-relate with its environment and figures out what would be the perfect possible outcome. The agent follows the notion of the trial and hit method. Here, the agent is then guerdoned with one point for one correct or penalized by one point for one incorrect answer, and based on the correct reward points gained by the agent, the model will train itself. Then once again, when the model is trained, the model would get prepared to make predictions regarding the fresh data given to it [1].

3. Applications of Machine Learning

Machine learning is probably the most exhilarating technology that one has come across. As obvious from its name, it makes a computer similar to a human being [3]. Machine learning is used extensively all around. Some of its applications are

- a) **Online Fraud Detection:** Machine learning has made our online transactions safe and secured by detecting fraud in our transactions. When we execute some online transactions, there may be certain ways in which a fraudulent type of transaction could take place for example fake accounts, fake ids, and stolen money in the process of a transaction. So to find this, Feed Forward Neural network aids us in checking if it is a genuine transaction or if it is a fraud transaction.
- b) **Spam detector:** When we get an email, it is filtered out automatically as important, spam, or normal. We receive an important email in the inbox with the important symbol attached and spam emails in our spam box. This is possible because of the technology behind it which is Machine learning.
- c) **Predictions:** If a person wants to go to a new place, he might take help of the Google Maps, which will direct us, the correct path with the shortest route and also predict the traffic conditions. It shows the traffics conditions for example whether traffic is clear or not, fast-moving, or congested with the aid of two different ways: Real-Time location of the automobile from the Google Map app and sensors or the Average time, which has been taken on the previous days at the same time.
- d) **Virtual Personal Assistants:** We have different virtual personal assistants such as Alexa and Siri. These assistants help us in finding the information using our instructions. Virtual assistants can also aid us in another way by using our voice instructions such as Play songs, call, send an email, Schedule an appointment, etc.
- e) **Image recognition:** Image recognition is probably the most common application using machine learning. It finds its usage in identifying certain objects, people, venues, and digital images, etc. The most common use of image recognition is Automatic friend tagging suggestion, Facebook provides this feature.

4. Steps for machine learning

The process for machine learning occurs in 4 important steps-

- a) **Gathering of data:** The first and foremost step is to feed some data to the computer through which our model learns. The data can be in any kind of formats such as text files or excel sheets.
- b) **Data preparation:** This step involves taking only the useful information from the whole data set which is fed into the computer. The data which is useful for the model for processing is only taken into consideration and the rest is discarded. This step also involves removing the unwanted data, checking for values that are missing, and treatment of the outliers.
- c) **Training set and testing set:** After that data is filtered in the second step, that data is then divided into 2 sets. One of the sets is called a training set which is used for creating the models and the second set, i.e. test sets is then used to check the accuracy of the model created [3].
- d) **Evaluation:** This step involves evaluating the model. To verify the accuracy of the created model, the model is tested on some data which was not present in the data used for its creation. Here, the test data is used.

5. Convolutional Neural Network

A Convolutional Neural Network also called ConvNet or CNN is a Deep Learning algorithm that takes into account an image as input and then assigns importance i.e. weights and biases to certain aspects or objects in the image and can differentiate them from the other [2-3]. The pre-processing required in a Convolutional neural network is lower as compared to the other classification algorithms available. Technically, deep learning models take each image as input and then pass it through several

convolution layers with filters (also called Kernals), Pooling layers, fully connected layers (FC) and then apply activation functions to classify the object with probable values between 0 and 1.

6. Steps for building a Convolutional neural network

- 1) Firstly, we provide an input image into the convolution layer
- 2) Next, we choose some parameters; apply filters with padding and strides, if required. We then perform convolution on the input image and apply activation functions [1].
- 3) We then need to perform pooling to reduce dimensionality size
- 4) Many convolution layers can be added as per need
- 5) Flatten the output and feed into a dense layer
- 6) Finally, train the model using the training set.

7. Dataset

In this project, I have used the CIFAR-10 dataset (Figure 1). This dataset consists of 60 thousand 32x32 RGB images with 10 classes, each class having 6000 images per class. There are 50 thousand training images and 10 thousand test images. This dataset was collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The classes are completely mutually exclusive. There is no overlap between the classes, automobiles, and trucks.

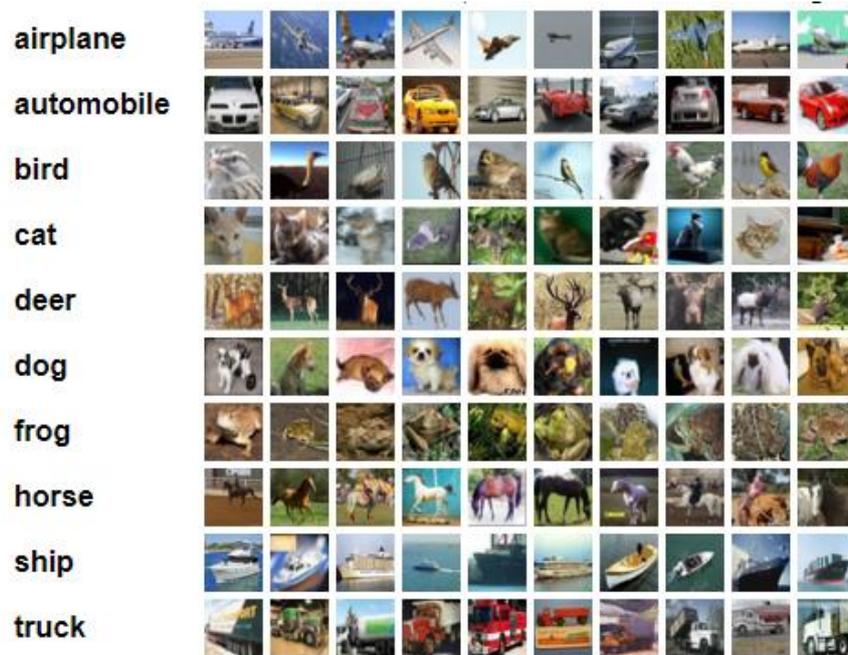


Figure 1. CIFAR-10 Dataset

8. Methodology

In this image classification task, I have used my knowledge of convolutional neural networks and, with the help of TensorFlow and Keras, have created a model to classify the images in the CIFAR-10 dataset using the jupyter notebook.

We first have imported some libraries necessary for this task.

- a) **TensorFlow:** TensorFlow is a Python library that is imported for fast numerical computing. It was created and released by Google. TensorFlow is a foundation library that can be used to build models of deep learning directly or by using some wrapper libraries that can simplify the process built on top of TensorFlow [14].
- b) **OS:** The OS library in Python provides useful functions for interacting with one's operating system. The OS module comes under Python's standard utility modules. This module provides a portable way of using the operating system's dependent functionality. The 'os' and 'os.path' modules subsume various functions to interact with the file system.
- c) **Numpy:** The Numpy library is the core library used for scientific computing in Python. This library provides a high-performance multidimensional array object and various useful tools for working with such arrays.
- d) **Matplotlib:** This useful library is a visualization library. One of the many benefits of visualization is that it allows us visual access to a large amount of data in easily digestible visuals such as line, bar, scatter, histogram, etc [14].

Next, I have created a function called `get_three_classes` to pre-process the CIFAR-10 dataset. For simplification, I have used only the first 3 classes in the dataset i.e. 'Airplane', 'Bird', and 'Automobiles'. After selecting the first three classes, it is necessary to remove the unwanted data i.e. the images for the other classes as they will be superfluous to our model. Using functions from the Numpy module, images corresponding to the 3 selected classes are selected and then are shuffled randomly and stored. Lastly, the labels are converted to their numerical representation using One-Hot encoding. Machines understand numbers and not text. Hence, we need to convert each text category i.e. our classes, to numbers for the machine to process them using mathematical equations.

The next step was to load the CIFAR-10 dataset from Keras. Deep neural networks, being the current passion, the convolution of its extensive frameworks have always been a hurdle for their application for all the fledglings to machine learning. Also, there were proposals for up-graded and simplified high-level API for building neural network models. Keras, one of the various prime high-level neural networks API, is written in Python language and aids many back-end neural network computation engines. On loading the dataset, the dataset was divided into the training set and test set. Using the `get_three_classes` function defined previously, we select the first classes and get their samples, and put them in the train and test set. On printing the summary of our train and test sets using the `shape` function, we see that our train set has 15,000 examples, 32x32 is the size of the images, and 3 means RGB i.e. the pictures are colored. Our test set has 3,000 examples.

Visualizing data can be helpful in many ways. Here, I have created another function `show_random_examples` to visualize my new data using the MatPlot library. After creating the model, this function will be used for displaying the predictions. Currently, I have displayed randomly any 10 images with their labels just to verify if up till now the code is working fine. Using the functions from the MatPlot library, this function would display those 10 random examples i.e. the images from the dataset and their labels either in green or red color (Figure 2). The green color will signify that the model classified the image correctly and the red color will signify that the model classified the image wrongly.





Figure 2. Visualizing Dataset

My next step was to create the model. In the first two lines, I simply imported different layers from Keras.

1. A **convolutional layer** is a layer where the required filters are applied to the original input.
2. A **pooling layer** is used to reduce the spatial size of the representation to reduce the number of parameters and computation required in the network.
3. **Batch normalization** is a technique used for improving the performance, speed, and stability of deep neural networks. As the name suggests, it is used to normalize the input layer by re-centering and re-scaling.
4. **Dropout** is simply a regularization technique. A simple way to avoid over-fitting. Using this, randomly selected neurons are ignored during training.
5. **Flattening** is used to transform a 2-dimensional array of features to a 1-D array that can be used as input to a fully connected (FC) neural network layer. It is used before the FC layer.
6. A **dense layer** is a basic deeply connect neural network layer. This layer operates below with the input and then returns one output.

$$\text{Output} = \text{activation}(\text{dot}(\text{input}, \text{kernel}) + \text{bias})$$

After importing the layers from Keras, I created two more functions called `add_conv_block` and `create_model`, using which the model is created. Then the summary of the model is displayed (Table 1).

Table 1. Summary of the model.

Layer (type)	Output Shape	Param #
Conv2d	(None, 32, 32, 32)	896
Batch_Normalization	(None, 32, 32, 32)	128
Conv2d_1	(None, 30, 30, 32)	9248
Max_Pooling2d	(None, 15, 15, 32)	0
Dropout	(None, 15, 15, 32)	0
Conv2d_2	(None, 15, 15, 64)	18496
Batch_Normalization_1	(None, 15, 15, 64)	256
Conv2d_3	(None, 13, 13, 64)	36928
Max_Pooling2d_1	(None, 6, 6, 64)	0
Dropout_1	(None, 6, 6, 64)	0
Conv2d_4	(None, 6, 6, 128)	73856
Batch_Normalization_2	(None, 6, 6, 128)	512
Conv2d_5	(None, 4, 4, 128)	147584
Max_Pooling2d_2	(None, 2, 2, 128)	0
Dropout_2	(None, 2, 2, 128)	0
Flatten	(None, 512)	0
Dense	(None, 3)	1539
Total Params: 289,443		
Trainable Params: 288,995		
Non-Trainable Params: 448		

After the creation of the model, I had to fit the model on the validation data using the validation set. For testing, I normalized the set. I set the epoch value to 10 and batch size to 128. I used EarlyStopping callback to stop the training when my monitored metric i.e. accuracy has stopped improving. The Patience parameter was set to 3 which means that after 3 numbers of epochs if there is no improvement in the val_accuracy, the training will be halted. In addition to the EarlyStopping callback, I also used ModelCheckpoint to save my model or weights in a different file at some interval, so the model or weights can be loaded. Later again, I saved my model in a separate file called 'models'.

After fitting my model, using the MatPlot library, I plotted a graph showing the validation data and training data with the val_accuracy and accuracy (Figure 3).

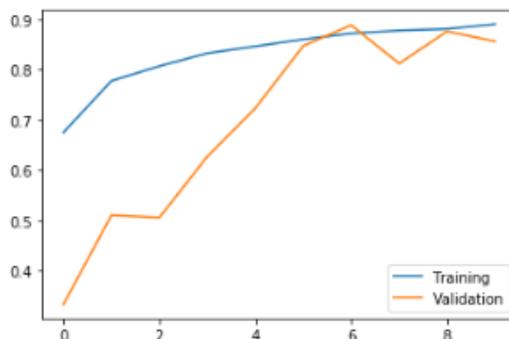


Figure 3. Graph of Validation data and Training data.

As the final step, I loaded the best model i.e. the one with the highest accuracy, and then using the predict function, my model made the predictions and with help of the show_random_examples, the predictions are displayed (Figure 4). Also, I displayed the model execution (Table 2) showing the epoch, time, loss, accuracy, validation loss, and validation accuracy.



Figure 4. Predictions

Table 2. Model execution

Epochs	Time	Loss	Accuracy	Val_loss	Val_Accuracy
Epoch 1/10 15000/15000	4ms/sample	0.8761	0.6753	2.1125	0.3333
Epoch 2/10 15000/15000	4ms/sample	0.5437	0.7777	1.9272	0.5103
Epoch 3/10 15000/15000	5ms/sample	0.4787	0.8071	1.6110	0.5057
Epoch 4/10 15000/15000	5ms/sample	0.4250	0.8327	0.8738	0.6273
Epoch 5/10 15000/15000	5ms/sample	0.3914	0.8465	0.7443	0.7233
Epoch 6/10					

15000/15000	5ms/sample	0.3652	0.8603	0.4366	0.8473
Epoch 7/10 15000/15000	4ms/sample	0.3292	0.8722	0.2902	0.8887
Epoch 8/10 15000/15000	4ms/sample	0.3199	0.8780	0.6152	0.8123
Epoch 9/10 15000/15000	4ms/sample	0.3096	0.8811	0.3315	0.8763
Epoch 10/10 15000/15000	5ms/sample	0.2856	0.8907	0.4041	0.8563

9. Conclusion

Machine learning is a rapidly developing field in Computer Engineering. It has applications in each other field of study and is as of now being actualized industrially because machine learning can take care of issues excessively troublesome or tedious for people to explain. An advantage to utilizing a convolutional neural system is that it is intended to all the more likely handle picture and speech recognition tasks. Rather than hidden layers, convolutional neural networks have a convolutional and pooling layer. It is a result of these layers that convolutional neural networks are favored for Image classification tasks.

Acknowledgment

First of all, I would like to thank Dr. Pawan Singh for steering me in the right direction in my beginnings. When I look at the amount of knowledge, I have gained to date in my B. Tech. journey, I realize how much time and effort spent toward the completion of this work; not only by me but also by key individuals whom I feel very indebted.

I gratefully acknowledge the support and guidance of my supervisor Prof. Dr. Pawan Singh. I would not imagine completing this work without all his advice. I acknowledge the time he spent in our weekly meetings, even with his busy schedule and other commitments. I am deeply indebted to him for his patience. He assisted in all aspects of this work by discussing new ideas for writing and completing this project.

I would like to thank Amity University, Lucknow Campus for providing me a wonderful platform for this work. I would also like to thank the library for the knowledge I acquired with its help. I am highly obliged to my HOD, Dr. Deepak Arora, and my HOI, Wg. Cdr. Anil Tiwari for helping me in the completion of this report. Lastly, I want to thank my family for supporting me entirely. I would like to express my deep gratitude and appropriation to my family and friends who were very cooperative and kind in getting me on the road.

References

- [1] M. Swamynathan, "in Mastering Machine Learning with Python in Six Steps," Berkeley, CA: Apress, pp. 297–344, 2017.
- [2] S. M. D. Lainez and D. B. Gonzales, "Automated fingerlings counting using convolutional neural network," in IEEE 4th International Conference on Computer and Communication Systems (ICCCS), pp.67-72, Feb 2019.
- [3] K. Narmada, G. Prabakaran, S. Mohan, "Classification and stage prediction of lung cancer using convolutional neural networks," vol. 8, no. 10, pp. 993–998, Aug 2019.
- [4] J. Zhang, F. Hu, L. Li, et al, "An adaptive mechanism to achieve learning rate dynamically," Neural Comput. Appl., vol. 31, no. 10, pp. 6685–6698, 2019.
- [5] H. A. Shiddieqy, F. I. Hariadi, and T. Adiono, "Implementation of deep-learning based image classification on single board computer," in International Symposium on Electronics and Smart Devices (ISESD), pp. 133-137, oct 2017.
- [6] S. Singh and N. Singh, "Object classification to analyze medical imaging data using deep learning," in International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), 201 pp. 1-4, March 2017.



- [7] K. He, X. Zhang, S. Ren, et al, "Deep residual learning for image recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, June 2016.
- [8] N Srivastava, G Hinton, A Krizhevsky, et al,"Dropout: a simple way to prevent neural networks from overfitting." The journal of machine learning research, vol.15, no.56, pp.1929-1958, 2014.
- [9] S. Srinivas, R. K. Sarvadevabhatla, K. R. Mopuri, N. Prabhu, S. S. S. Kruthiventi, and R. V. Babu, "A taxonomy of deep convolutional neural nets for computer vision," Front. Robot. AI, vol. 2, no.36, pp.1-15, Jan 2016.
- [10] A. Krizhevsky, G. Hinton," Learning multiple layers of features from tiny images." Technical Report TR-2009, University of Toronto, Toronto, April 2009.
- [11] Y.Yang & T.M. Hospedales, "Deep neural networks for sketch recognition", 2015.
- [12] D. Ciregan, U. Meier and J. Schmidhuber, "Multi-column deep neural networks for image classification," in IEEE Conference on Computer Vision and Pattern Recognition, pp.3642-3649, June 2012.
- [13] D. C. Cireş, U. Meier, J. Masci, et al," Flexible, High Performance Convolutional Neural Networks for Image Classification," IJCAI, pp. 1237-1242 ,2011.
- [14] K. Chauhan, S. Ram "Image Classification with Deep Learning and Comparison between Different Convolutional Neural Network Structures using Tensorflow and Keras," International Journal of Advance Engineering and Research Development, vol. 5, no.2, pp. 533-538, 2018.
- [15] S. Khan and S.-P. Yong, "A deep learning architecture for classifying medical images of anatomy object," in 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 1661-1668, Dec 2017.

