International Journal for Modern Trends in Science and Technology, 6(12): 235-239, 2020

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST061243
Available online at: http://www.ijmtst.com/vol6issue12.html

Comparative Analysis of Machine Learning Algorithms with and without Feature Extraction

Vatsal Gupta[1] | Saurabh Gautam[2]

[1]B-tech. I.T., Maharaja Agrasen Institute of Technology, GGSIPU, New Delhi, India
[2]Assistant Professor I.T., Maharaja Agrasen Institute of Technology, GGSIPU, New Delhi, India

To Cite this Article
Vatsal Gupta and Saurabh Gautam, "Comparative Analysis of Machine Learning Algorithms with and without Feature Extraction", *International Journal for Modern Trends in Science and Technology*, 6(12): 235-239, 2020.

Article Info
Received on 10-November-2020, Revised on 02-December-2020, Accepted on 06-December-2020, Published on 10-December-2020.

ABSTRACT

Image recognition is one of the core disciplines in Computer Vision. It is one of the most widely researched topics of the last few decades. Many advances in image recognition in the past decade, has made it one of the most efficient and powerful disciplines of all, having its applications in every sector including Finance, Healthcare, Security services, Agriculture and many more. Feature extraction is an integral part of image recognition. It helps in training the model more efficiently and with a higher accuracy, by getting rid of any unwanted or unnecessary features, thus reducing the dimensionality of the input image. This also helps in reducing the computational resources required by the algorithm to train, thus making it affordable for people with low end setups. Here we compare the accuracies of different machine learning classification algorithms, and their training times, with and without using feature Extraction. For the purpose of extracting features, a convolutional neural network was used. The model was trained and tested on the data of 12 classes containing a total of 2,175 images. For comparisons, we chose the Logistic regression, K-Nearest Neighbors Classifier, Random forest Classifier, and Support Vector Machine Classifier.

KEYWORDS: *Feature Extraction, Image Classification, Machine Learning, Accuracy Metrics, MobileNetV2*

I. INTRODUCTION

Computer vision is a multi-disciplinary field in Computer Science, that deals with the way the computers extract and manipulate the information in digital images and videos. In simpler terms, it tries to understand and automate the tasks the human visual system does [1]. It has become one of the most noteworthy and exploited technology of this decade, which many people believe would continue for a significant amount of time. With the passage of time, it is becoming more reliable and convenient. All of its applications in various sectors, comes down to its core, i.e. extracting information from digitally stored visual data.

The most well-known and widespread task of computer vision is Image Classification. It uses a predefined set of images loaded into the system to categorize the given input image and classify it according to the results. In other words, it is the process of taking as an input, an image and generating its corresponding class label or probability that the image is a particular class.

The foundation of using Convolutional Neural Networks (CNN) for feature extraction was laid by Yann LeCun in 1989, when he used handwritten digits recognition to read zip codes on envelopes and digits on checks [2].

Despite their ingenuity, due to high computational resources required, they couldn't be scaled, thus remained on the sidelines of computer vision and artificial intelligence. However, with exponential increase in amount of data in the early

21st century, Alex Krizhevsky proposed a CNN-based solution [3] for the ImageNet Large Scale Visual Recognition Challenge, which improved theability of CNNs to capture and represent complex features. Using GPU training and ReLU non-linearity, he was able to greatly reduce high computation cost.

In this paper, we present a detailed study of effect of using CNNs for feature extraction by contrasting the performance of various classification algorithms (Logistic Regression, Decision Trees, Random Forests and k-NN) with and without the use of features extracted and comparing the time consumed in the two situations.

II. Related Theory and Fundamentals

A. Convolutional Neural Networks as Features extractors

The elementary unit of CNNs is the Convolutional Layer. Convolution is an advanced mathematical operation that is used to merge two sets of data or information. Here, the sets are the input image and the convolution filter, also called the kernel.

*The Convolution Operation:*The convolution operation is performed by sliding the filter over the input image. At every location, element-wise matrix-multiplication takes place, followed by the some of result, as illustrated in figure 1. This produces a Feature Map. We perform multiple such operations on the input image, each using a different filter, thus resulting in a different feature map. Then, we get the final output of the convolution layer, by stacking all these feature maps together.

In Figure 1, the sum of values in the green highlighted square is 4, which forms the first value in the feature map. Rest of the values are calculated by traversing the green kernel towards the right and bottom respectively.

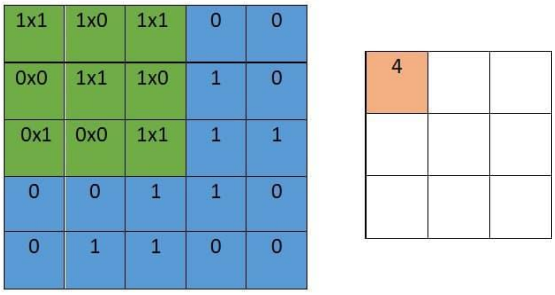

Fig. 1. Producing Feature Map from Input image and filter

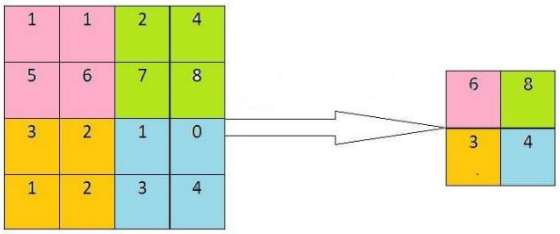

Fig. 2. Dimensionality Reduction by Max Pooling, with kernel size of 2

These feature maps depict different features in an image (such as vertical and horizontal lines, curved edges, color contrasts etc.) that would be helpful while classifying these images.

The Pooling Operation: Usually after a convolution operation, a pooling operation is performed, to reduce the dimensionality. This reduces the number of parameters, which in turn reduces the training time and computational resources required, and simultaneously combats overfitting.

In Figure 2, the max-pooling operation, with a kernel size of 2, chooses the maximum value in the kernel, and outputs it at the corresponding position. The kernel traverses towards the right and bottom respectively, with a stride equal to the kernel size.

B. Linear Classifiers

Linear Models have been studied extensively in the last few decades. They make a prediction using a linear function of the input features. It is a frontier that best segregates the two classes with a hyper-plane or a line. It performs best in case of extreme cases. They are highly effective in higher dimensional spaces, especially if the number of dimensions is greater than the number of examples.

C. Support Vector Machines

In Support Vector Machines (SVM) algorithm, each data item is plotted in n-dimensional space, where n represents the number of features, with value of each feature being the value of the particular coordinate. Then by finding the hyper-plane, differentiation between the two classes takes place. This is generalized to one vs rest approach from the one vs one approach, to incorporate multiclass classification problem.

D. K-Nearest Neighbors Classifier

The K-Nearest Neighbor (K-NN) algorithm is the simplest machine learning algorithm of all. To build the model, one only has to store the training data. For making a prediction for a new data point, the algorithm finds the closest data points in the training dataset, based on similarity measures (such as distance function), or its neighbors.

E. Random Forest Classifier

The Random Forest classifier consists of a large number of individual decision trees, that operate together as an ensemble. Each individual tree, in the random forest, generates a class prediction. The predicted class is the mode of all the individual tree's predictions. Here, a large number of uncorrelated trees operate together, in order to outperform any of the individual trees.

III. RELATED WORK

In paper [5], the authors, Hui-huang Zhao & Han Liu proposed a framework, involving feature extraction using CNNs and a multi-level fusion of diverse classifiers. By preparing different feature sets from Handwritten Digits MNIST Dataset and using different learning algorithms for classifiers' training, they have designed to increase the diversity among classifiers.

In paper [6], the authors compare the accuracies obtained on the CIFAR-10 dataset, using CNN (for feature extraction) and various machine learning algorithms (such as K-Nearest Neighbor, Support Vector Machine Classifier and Fully-connected neural network classifier).

In paper [7], the authors used computer vision to detect pneumonia from the frontal chest X-rays. They used CNN's feature extraction to extract the features from theX-rays, followed by classifying them as normal and abnormal chest X-rays with the help of support vector machines.

IV. DATASET

The dataset used is obtained from Kaggle, which consists of 17534 images of 105 different celebrities. Here only a subset of this dataset containing randomly chosen 12 celebrities is used. The dataset contains cropped pictures of these celebrities collected from Pinterest, having varying dimensions. The dataset was further divided into training and test set for better performance. The algorithm is trained over the training set, while the accuracy is tested on the test set.

V. ACCURACY METRICS

Evaluating the quality of the machine learning model is extremely important for continuing to improve it until it performs as best as it can. In case of classification problems, the evaluation metrics compare the expected class label to the predicted class.

A. Confusion Matrix

A Confusion Matrix is a table containing four different combinations of actual values and predicted values. This helps in visualizing different outputs and calculate the Precision, Recall, Accuracy, and F-1 Score.

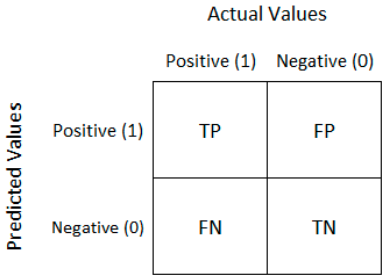

Fig. 3. Confusion Matrix

- True Positives (TP)- It represents the number of times that the proposed model predicted the value YES when the actual output was also YES.
- True Negatives (TN)- It represents the number of times our model predicted NO and the actual output was also NO.
- False Positives (FP)- It represents the number of times our model predicted YES but the actual output was NO. This is also known as Type-1 Error.
- False Negatives (FN)- It represents the number of times our model predicted NO but the actual output was YES. This is also known as Type-2 Error.

B. Precision

It represents the fraction of relevant instances among he retrieved instances. It is basically the ratio of true positives to the sum of true positives and false positives. It expresses the proportion of data points our model says was relevant actually were relevant.

C. Recall

It expresses the instances relevant in the dataset. It is also known as sensitivity.

Because of their inverse relationship, it is important to examine both the recall and precision.

D. Accuracy

It represents the number of correct classifications with respect to the total number of classifications.

E. F1-Score

The F1-score is the measure of accuracy on the testing data as a function of precision and recall. It expresses the number of instances the model classifies correctly without missing a significant number of instances. It can have a maximum value of one and a minimum value of zero.

F1 = 2 * (precision * recall) / (precision + recall)

Model	Time Consumed (sec)	Accuracy	Precision	Recall	F1-Score
Logistic Regression	2.99	0.8791	0.87	0.87	0.87
K-Nearest Neighbor Classifier	0.391	0.8604	0.86	0.85	0.85
Random Forest Classifier	1.65	0.8744	0.87	0.87	0.87
Support Vector Machines Classifier	1.41	0.879	0.88	0.87	0.87

Table 1. Classification Algorithms and their evaluations with feature extraction

Model	Time Consumed (sec)	Accuracy	Precision	Recall	F1-Score
Logistic Regression	668.92	0.3674	0.36	0.35	0.35
K-Nearest Neighbor Classifier	24.06	0.2627	0.29	0.25	0.24
Random Forest Classifier	25.12	0.3465	0.31	0.32	0.31
Support Vector Machines Classifier	334.25	0.4012	0.41	0.38	0.38

Table 2. Classification Algorithms and their evaluations without feature extraction

VI. Experimental Setup And Methodology

The dataset used was only a subset of the entire dataset, containing only 12 classes out of 105. The dataset contains 2,175 which are required to be divided into training and testing sets for training the network and correspondingly evaluating the performance of the network respectively.

For the CNN for feature extraction, we used MobilNetV2 network [8]. The architecture of this system resembles an inverted residual structure, where both the input and output of the residual block are extremely thin bottleneck layers [8]. This is in contrast to the traditional residual models, using expanded representations in the input. MobileNetV2 uses lightweight depth wise convolutions to filter features in immediate expansion layer.[8]

To obtain the features, the images were passed into the MobileNetV2 network, and the corresponding predictions were used as the inputs for the different machine learning algorithms. These algorithms were then trained on these features with corresponding labels.

Thereafter, the same set of algorithms, were trained on the original images and labels. The performance of the different algorithms for different types of inputs was then compared for each algorithm.

VII. RESULTS

The MobileNetV2 network was trained on the training dataset as the CNN responsible for feature extraction. Another model, was then created, having the same images as inputs. However, it generated the features instead of the class, by outputting the outputs from an intermediate layer. During the training of the network, an accuracy of 85.11% and a loss of 0.616 was obtained.

These features were then used to train Logistic Regression Classifier, Random Forest Classifier, Support Vector Machines Classifier, and K-Nearest Neighbors Classifier. Table 1 shows the results obtained and the time consumed by each algorithm to undergo training.

Table 1 depicts performance of various classification algorithms with feature extraction,and the time consumed (in seconds) for training the models. The K-NN classifier is the most basic of all requiring the least amount of time when compared with the rest, while giving a respectable score. A slightly higher time consumption in SVM and Logistic Regression classifier can be explained by their approach of linear boundaries between the classes.

Table 2 depicts performance metrics for various classification algorithms without any kind of feature extraction, and the time consumed (in seconds) for training the models. When compared with feature extracted results, the time consumed have grown drastically for the models trained on original images for logistic regression and support vector machines. K-Nearest Neighbor classifier and Random forest, on the other hand, does not have such a drastic increase in time consumed. This may be explained by a correspondingly large number of features present in the original images. The ability of Random Forest to handle large amounts of features, and simplicity of K-Nearest Neighbors algorithm, allow them to consume much shorter span of time for training.

The accuracies of the models, also drop drastically when switching from feature extraction to original images, as due to a very large number of

features, generalization for better accuracy is not possible. With original images, presence of large numbers of unnecessary features, also makes the models more complex, thus decreasing the overall accuracy.

The models showcase a larger variance in different metrics amongst each other when trained on original images, as compared to when they're trained on extracted features respectively. This may a shortcoming in various models' ability to handle unnecessary features.

VIII. CONCLUSION

We have contrasted the two approaches that could be used for image recognition. One, that uses machine learning algorithms directly on original images without any kind of feature extraction, and the other wherein we first extracted the features and trained the machine learning algorithms on these extracted features. The former required a large amount of time, and gave a low accuracy on the testing data, while the later consumed lesser amount of time, while maintaining a high accuracy, and giving a respectable score of other metrics as well. Such a contrast could be due to, large number of unnecessary features presented in original images that impacted the generalization of the model. For future work, other feature extraction techniques could be implemented to improve the performance of the model.

REFERENCES

[1] Dana H. Ballard; Christopher M. Brown (1982). Computer Vision. Prentice Hall. ISBN 978-0-13-165316-0.

[2] 2. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, Backpropagation Applied to Handwritten Zip Code Recognition; AT&T Bell Laboratories

[3] 3. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60, 6 (June 2017), 84–90. DOI: https://doi.org/10.1145/3065386

[4] D. Varshni, K. Thakral, L. Agarwal, R. Nijhawan and A. Mittal, "Pneumonia Detection Using CNN based Feature Extraction," 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, 2019, pp. 1-7, doi: 10.1109/ICECCT.2019.8869364.

[5] Zhao, Hh., Liu, H. Multiple classifiers fusion and CNN feature extraction for handwritten digits recognition. Granul. Comput. 5, 411–418 (2020). https://doi.org/10.1007/s41066-019-00158-6

[6] Zhao, Hh., Liu, H. Multiple classifiers fusion and CNN feature extraction for handwritten digits recognition. Granul. Comput. 5, 411–418 (2020). https://doi.org/10.1007/s41066-019-00158-6

[7] Zhao, Hh., Liu, H. Multiple classifiers fusion and CNN feature extraction for handwritten digits recognition. Granul. Comput. 5, 411–418 (2020). https://doi.org/10.1007/s41066-019-00158-6

[8] Zhao, Hh., Liu, H. Multiple classifiers fusion and CNN feature extraction for handwritten digits recognition. Granul. Comput. 5, 411–418 (2020). https://doi.org/10.1007/s41066-019-00158-6