ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 7, Issue 5, May 2020

Fruit Recognition Using Machine Learning

[1] Arjun Paliwal, [2] Sudhanshu Gaur, [3] Shwetank Tripathi, [4]UtkarshKumar Srivastava
[1][2][3][4] IMSEC, AKTU University, Ghaziabad, Uttar Pradesh,India

Abstract: In this paper we proposed a predictive model to develop which recognizes the fruits to replace the manual recognition system.Model is trained on a dataset which contains features mass ,colour ,size, height, width.There are four classification algorithmsnamely Naive Bayes, K Nearest Neighbour, Decision Tree and Logistic Regression which are used in this experiment. The performances of all the algorithms are evaluated on measure accuracy. Results obtained shows that K-Nearest Neighbour outperforms the best on the measure accuracy.

1. INTRODUCTION

The aim of this paper is to propose a new dataset containing dimensions of popular fruits such as mass, width, height and colour score. The dataset can be downloaded from the Kaggle. Kaggle allows users to find and publish data sets, explore and build models in a web-based data-science environment, work with the other data scientists, and machine learning engineers, and enter the competitions to solve the data science challenges. The reader is encouraged to access the latest version of the dataset from the above indicated addresses.

India is an agriculture country. All the pre-harvest and post-harvest process are done manually with help of labour. The post-harvest process includes sorting, grading and recognition of fruits.

Automation is playing important role in day today life. Their main source of income is agriculture. Exporting of fresh fruit is increased day to day from India. People are very conscious about their health; they prefer only fresh, good quality fruit.

The part of a more complex project that has the target of obtaining a classifier that can identify a much wider array of objects from images. This fits the current trend of companies working in the augmented reality field. During its annual I/O conference, Google announced that is working on an application named Google Lens which will tell the user many useful information about the object toward which the phone camera is pointing. First step in creating such application is to correctly identify the objects. The software has been released later in 2017 as a feature of Google Assistant and Google Photos apps. As the start of this project we chose the task of identifying fruits for several reasons. As we know that India is an agriculture country. All the pre-harvest and post-harvest process are done manually with help of labour. The post-harvest process includes sorting, grading and recognition of fruits.

Thus we want to see how well can an artificial intelligence complete the task of classifying them.

The paper is structured as follows :- in the first part we will shortly discuss a few outstanding achievements obtained using machine learning for fruits recognition. In the second part we will describe the training and testing data used as well as the obtained performance. Finally, we will conclude with a few plans on how to improve the results of this project.

2 RELATED WORK

In this section we review several previous attempts to use machine learning for fruits recognition. A method for recognizing and counting fruits from images in cluttered greenhouses is presented.

The targeted plants are peppers with fruits of complex shapes and varying colors similar to the plant canopy. The aim of the application is to locate and count green and red pepper fruits on large, dense pepper plants growing in a greenhouse.

The training and validation data used in this paper consists of 28000 images of over 1000 plants and their fruits. The used method to locate and count the peppers is two-step: in the first step, the fruits are located in a single image and in a second step multiple views are combined to increase the detection rate of the fruits.

The approach to find the pepper fruits in a single image is based on a combination of finding points of interest, applying a complex high-dimensional feature descriptor of a patch around the point of interest and using a so-called bag-of-words for classifying the patch. It presents a novel approach for detecting fruits from images using Machine Learning. For this purpose the authors adapt a Faster Region-based convolutional network. The result is a multi model network which obtains much better performance than the existing networks.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 7, Issue 5, May 2020

3 MACHINE LEARNING

ALGORITHM USED:-
Decision tree
K-nearest neighbour
Naive bayes
Logistic Regression

Machine learning algorithms that use multiple layers that contain nonlinear processing units. Each layer uses the output from the previous layer as input. Machine learning algorithms use more layers than shallow learning algorithms. Another Machine learning algorithm is the recursive neural network. In this kind of architecture the same set of weights is recursively applied over some data. Recurrent networks have shown good results in natural language processing. Yet another model that is part of the Machine learning algorithms is the deep belief network. A deep belief network is a probabilistic model composed by multiple layers of hidden units. The usages of a deep belief network are the same as the other presented networks but can also be used to pre-train a deep neural network in order to improve the initial values of the weights. This process is important because it can improve the quality of the network and can reduce training times. Deep belief networks can be combined with convolutional ones in order to obtain convolutional deep belief networks which exploit the advantages offered by both types of architectures. In the area of image recognition and classification. This served as one of the reasons we chose to use a deep neural network in order to identify fruits from images. Deep neural networks have managed to outperform other machine learning algorithms. They also achieved the first superhuman pattern recognition in certain domains. This is further reinforced by the fact that deep learning is considered as an important step towards obtaining Strong AI. Secondly, deep neural networks – specifically convolutional neural networks – have been proved to obtain great results in the field of image recognition. We will present a few results on popular datasets and the used methods.

3.1 Decision tree Algorithm-

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too.

The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).

In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

Decision Tree

```
In [25]: from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier().fit(X_train, y_train)
print('Accuracy of Decision Tree classifier on training set: {:.2f}'
     .format(logreg.score(X_train, y_train)))
print('Accuracy of Decision Tree classifier on test set: {:.2f}'
     .format(logreg.score(X_test, y_test)))

Accuracy of Decision Tree classifier on training set: 0.59
Accuracy of Decision Tree classifier on test set: 0.40
```

We can clearly see that the accuracy of the Decision Tree Classifer on Training set: 0.59

We can clearly see that the accuracy of the Decision Tree Classifer on Test set: 0.40

3.2 K-nearest neighbour Algorithm-

The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.

KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).

In the case of classification and regression, we saw that choosing the right K for our data is done by trying several Ks and picking the one that works best.

K-Nearest Neigbors ¶

```
In [26]: from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier()
knn.fit(X_train, y_train)
print('Accuracy of K-NN classifier on training set: {:.2f}'
     .format(knn.score(X_train, y_train)))
print('Accuracy of K-NN classifier on test set: {:.2f}'
     .format(knn.score(X_test, y_test)))

Accuracy of K-NN classifier on training set: 0.93
Accuracy of K-NN classifier on test set: 0.87
```

We can clearly see that the accuracy of the K-NN Classifer on Training set: 0.93

We can clearly see that the accuracy of the K-NN Classifer on Test set: 0.87

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 7, Issue 5, May 2020

3.3 Naïve Bayes Algorithm-

In machine learning, naïve Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes theorem with strong (naïve) independence assumptions between the features. They are among the simplest Bayesian network models. But they could be coupled with Kernel density estimation and achieve higher accuracy levels.

Naïve Bayes has been studied extensively since the 1960s. It was introduced (though not under that name) into the text retrieval community in the early 1960s, and remains a popular (baseline) method for text categorization, the problem of judging documents as belonging to one category or the other (document categorization)(such as spam or legitimate sports or politics, etc.) with word frequencies as the features. With appropriate pre-processing, it is competitive in this domain with more advanced methods including support vector machines. It also finds application in automatic medical diagnosis.

Gaussian Naive Bayes

```
In [27]: from sklearn.naive_bayes import GaussianNB

         gnb = GaussianNB()
         gnb.fit(X_train, y_train)
         print('Accuracy of GNB classifier on training set: {:.2f}'
              .format(knn.score(X_train, y_train)))
         print('Accuracy of GNB classifier on test set: {:.2f}'
              .format(knn.score(X_test, y_test)))

Accuracy of GNB classifier on training set: 0.93
Accuracy of GNB classifier on test set: 0.87
```

We can clearly see that the accuracy of the GNB Classifer on Training set: 0.93

We can clearly see that the accuracy of the GNB Classifer on Test set: 0.87

3.4 Logistic Regression Algorithm-

Regression analysis is the process of estimating the relationships among variables and predicting where a particular variable belongs to which class. Regression is basically a two-class classification method. It is broadly of two types:

Linear and non-linear regression. Linear regression analysis has a classifier which is

represented by a straight line while non-linear regression analysis has a classifier represented by a curve.

Logistic regression is another technique borrowed by machine learning from the field of statistics.

Logistic Regression

```
In [24]: from sklearn.linear_model import LogisticRegression
         logreg = LogisticRegression()
         logreg.fit(X_train, y_train)

         print('Accuracy of Logistic regression classifier on training set: {:.2f}'
              .format(logreg.score(X_train, y_train)))
         print('Accuracy of Logistic regression classifier on test set: {:.2f}'
              .format(logreg.score(X_test, y_test)))

Accuracy of Logistic regression classifier on training set: 0.59
Accuracy of Logistic regression classifier on test set: 0.40
```

4

We can clearly see that the accuracy of the Logistic Regression Classifer on Training set: 0.59

We can clearly see that the accuracy of the Logistic Regression Classifer on Test set: 0.40

4 FRUITS-360 DATA SET

In this section we describe how the data set was created and what it contains

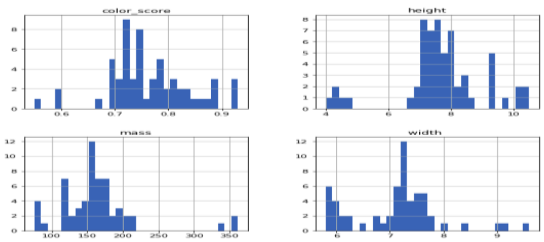
```
fruit_label,fruit_name,fruit_subtype,mass,width ,height,color_score
1,apple,granny_smith,192,8.4,7.3,0.55
1,apple,granny_smith,180,8,6.8,0.59
1,apple,granny_smith,176,7.4,7.2,0.6
2,mandarin,mandarin,86,6.2,4.7,0.8
2,mandarin,mandarin,84,6,4.6,0.79
2,mandarin,mandarin,80,5.8,4.3,0.77
2,mandarin,mandarin,80,5.9,4.3,0.81
2,mandarin,mandarin,76,5.8,4,0.81
1,apple,braeburn,178,7.1,7.8,0.92
1,apple,braeburn,172,7.4,7,0.89
1,apple,braeburn,166,6.9,7.3,0.93
1,apple,braeburn,172,7.1,7.6,0.92
1,apple,braeburn,154,7,7.1,0.88
1,apple,golden_delicious,164,7.3,7.7,0.7
1,apple,golden_delicious,152,7.6,7.3,0.69
1,apple,golden_delicious,156,7.7,7.1,0.69
1,apple,golden_delicious,156,7.6,7.5,0.67
1,apple,golden_delicious,168,7.5,7.6,0.73
1,apple,cripps_pink,162,7.5,7.1,0.83
1,apple,cripps_pink,162,7.4,7.2,0.85
1,apple,cripps_pink,160,7.5,7.5,0.86
1,apple,cripps_pink,156,7.4,7.4,0.84
1,apple,cripps_pink,140,7.3,7.1,0.87
1,apple,cripps_pink,170,7.6,7.9,0.88
3,orange,spanish_jumbo,342,9,9.4,0.75
3,orange,spanish_jumbo,356,9.2,9.2,0.75
3,orange,spanish_jumbo,362,9.6,9.2,0.74
3,orange,selected_seconds,204,7.5,9.2,0.77
3,orange,selected_seconds,140,6.7,7.1,0.72
3,orange,selected_seconds,160,7,7.4,0.81
3,orange,selected_seconds,158,7.1,7.5,0.79
3,orange,selected_seconds,210,7.8,8,0.82
3,orange,selected_seconds,164,7.2,7,0.8
3,orange,turkey_navel,190,7.5,8.1,0.74
3,orange,turkey_navel,142,7.6,7.8,0.75
3,orange,turkey_navel,150,7.1,7.9,0.75
3,orange,turkey_navel,160,7.1,7.6,0.76
3,orange,turkey_navel,154,7.3,7.3,0.79
3,orange,turkey_navel,158,7.2,7.8,0.77
3,orange,turkey_navel,144,6.8,7.4,0.75
3,orange,turkey_navel,154,7.1,7.5,0.78
3,orange,turkey_navel,180,7.6,8.2,0.79
3,orange,turkey_navel,154,7.2,7.2,0.82
4,lemon,spanish_belsan,194,7.2,10.3,0.7
4,lemon,spanish_belsan,200,7.3,10.5,0.72
4,lemon,spanish_belsan,186,7.2,9.2,0.72
4,lemon,spanish_belsan,216,7.3,10.2,0.71
4,lemon,spanish_belsan,196,7.3,9.7,0.72
```

IFERP
connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering
(IJERCSE)
Vol 7, Issue 5, May 2020

5 PERFORMANCE MEASURE



Out[14]:

	fruit_label	mass	width	height	color_score
count	59.000000	59.000000	59.000000	59.000000	59.000000
mean	2.542373	163.118644	7.105085	7.693220	0.762881
std	1.208048	55.018832	0.816938	1.361017	0.076857
min	1.000000	76.000000	5.800000	4.000000	0.550000
25%	1.000000	140.000000	6.600000	7.200000	0.720000
50%	3.000000	158.000000	7.200000	7.600000	0.750000
75%	4.000000	177.000000	7.500000	8.200000	0.810000
max	4.000000	362.000000	9.600000	10.500000	0.930000

6 FUTURE SCOPE

Further design can be modified by increasing size of conveyor belt so that it is possible to perform quality of fruit ,and increase accuracy of the system so that it can differentiate between artificial , hybrid colour from original fruit colour.

7 CONCLUSION

This test identify Fruits from the given dataset with the help of features of mass, colour score etc. present in that fruit are considered for sorting and grading of fruits.

This project tries to set up a start to an area that is less explored at the current time. During this project we were able to explore part of the deep learning algorithms and discover strengths and weaknesses. We gained knowledge on deep learning and we obtained a software that can recognize fruits from images. We hope that the results and methods presented in this paper can be further expanded in a bigger project.

References

[1] S. Arivazhagan, N. Shebiah, S. Nidhyanandhan, L.Ganesan, Fruit recognition using color and texture features, Journal of Emerging Trends in Computing and Information Sciences, 1, 2 (2010) 90–94. ⇒29

[2] S. Bargoti, J. Underwood, Deep fruit detection in orchards, IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 3626–3633. ⇒28

[3] R. Barth, J. Ijsselmuiden, J. Hemming, E. Van Henten, Data synthesis methods for semantic segmentation in agriculture: A Capsicum annuum dataset, Computers and Electronics in Agriculture 144 (2018) 284–296. ⇒29

[4] T. F. Chan, L. Vese, Active contours without edges. IEEE Trans. Image Process 10, (2001) 266–277. ⇒29

[5] H. Cheng, L. Damerow, Y. Sun, M. Blanke, Early yield prediction using image analysis of apple fruit and tree canopy features with neural networks, Journal of Imaging, 3, 1 (2017) 6. ⇒29

[6] D. C. Cire¸san, U. Meier, J. Schmidhuber, Multi-column deep neural networks for image classification, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Providence, pp. 3642–3649, 2012. ⇒30

[7] D. Clevert, T. Unterthiner, S. Hochreiter, Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs) CoRR abs/1511.07289, 2015. ⇒40

[8] J. Hemming, J. Ruizendaal, J. W. Hofstee. E J. Van Henten, Fruit detectability analysis for different camera positions in sweet-pepper Sensors 14, 4 (2014) 6032–6044. ⇒29

[9] K. Kapach, E. Barnea, R. Mairon, Y. Edan, O. Ben-Shahar, Computer vision for fruit harvesting robots state of the art and challenges ahead, Journal of Imaging 3, 1 (2017) 4–34. ⇒29

[10] M. Liang, X. Hu, Recurrent convolutional neural network for object recognition, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Boston. 2015. pp. 3367–3375. ⇒31

[11] D. Mumford, J. Shah, Optimal approximations by piecewise smooth functions and associated variational problems, Commun. Pure Appl. Math. 42 (1989) 577-685. ⇒29

[12] P. Ninawe, S. Pandey, A completion on fruit recognition system using knearestneighbors algorithm, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) 3, 7 (2014) 2352–2356. ⇒ 29

[13] S. Puttemans, Y. Vanbrabant, L. Tits, T. Goedem, Automated visual fruit detection for harvest estimation and robotic harvesting, Sixth International Conference on Image Processing Theory, Tools and Applications, 2016. ⇒29

[14] M, Rahnemoonfar, C. Sheppard, Deep count: fruit counting based on deep simulated learning, Sensors 17, 4

(2017) 905. ⇒29

[15] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Advances in Neural Information Processing Systems, 2015, 91–99. ⇒29