International Journal of Research in Engineering, Science and Management
Volume 5, Issue 4, April 2022
https://www.ijresm.com | ISSN (Online): 2581-5792

Determination of Recipe by Analysis of Various Ingredients

C. Nandini[1], Adrita Choudhury[2], A. N. Shreyas[3*], Harsh Jain[4], Hrithik Goyal[5]

[1]*Professor & HoD, Department of Computer Science and Engineering, Dayananda Sagar Academy of Technology and Management, Bengaluru, India*
[2,3,4,5]*Student, Department of Computer Science and Engineering, Dayananda Sagar Academy of Technology and Management, Bengaluru, India*

***Abstract*: Consumption of junk food can hold extreme levels of type of fatty acid, instance. At the same time, it concedes possibility have little digestive content in conditions of vitamins, mineral and fiber. Junk food consistently holds unsound preservatives and additional chemical compound. Sometimes family use junk food as equivalent accompanying inexpensive food prepared and served quickly. Unhealthful food frequently holds preservatives and flavorings that have an unfavorable effect on our fitness. According to the current mathematical studies, on an average about individual in four family eat inexpensive food prepared and served quickly per epoch. As orderly devouring of inexpensive food prepared and served quickly is sick, our model is delineated to benefit the children to entertain in style themselves, in their active-due often growth. To overcome these above issues, we need to evolve a structure that will smart to find the Recipe in which the additives captured as recommendation, is analyzed by way of dataset and the appropriate platters is urged to the consumer. In projected whole we are requesting Machine Learning Algorithm search out analyze the dataset, Ingredients are Determination of Recipe by Analysis of Various Ingredients likely in the form of manual or a countenance. It uses Image Processing method to recognize the pieces likely as a figure recommendation.**

***Keywords*: recipe determination, machine learning algorithm, knn, clustering, image processing, keras, django, python.**

1. Introduction

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks.

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

Food is an essential component of our individual and social interactions. Eating practices have direct impact on our fitness and well-being, while factors like, flavors and cooking formulas shape distinguishing cuisines that are few our personal and composite enlightening identities. Recent concerning details advances such as smartphones outfitted accompanying cameras and additional rich sensors, extensive networks and artificial intelligence have stimulated new uses of electronics connected with feed. For instance, common fare logging for diet listening demands knowledge and effort from the consumer, and is compulsive inaccuracies and overlooking. In contrast, an automatic meal glossary plan could act mechanical reasoning, annotation and record accompanying minimum human attack.

For instance, photos from smartphones are nearby yet strong introduction points to many applications including acknowledgment, retrieval or approval. In this direction, meal-oriented public networks and coffee shop review services have bloomed, place snack enthusiasts (like, foodies, gourmets, cooks) combine and share information (for instance, formulas, photos, comments about restaurants). The reasoning concerning this user provided data also supports appealing insight to comprehend consuming habits, cuisines and breeding's. This composite knowledge can still, in proper sequence, be leveraged by recognition models to upgrade their veracity. Thus, reliable drink reasoning from images is essential for these uses. Despite unusual advances in computer apparition, feed recognition in shrubs still debris a very challenging Determination of Recipe by Analysis of Various Ingredients question even for persons. We largely depend circumstantial and prior facts. Similarly, framework and prior knowledge maybe joined in automatic feed study systems. We review few current works in this place arising direction. According to the recent statistical studies, on an average about 1 in 4 people consume fast food per day. As regular consumption of fast food is unhealthy, our model is defined to benefit the youngsters to cook for themselves, in their busy-scheduled daily life. In Image Processing, an image is digitized, by performing operations on it, which results in an improved image to retrieve meaningful data. In computer science, digital image processing is the use of a digital computer to process digital images through an algorithm.

As a subcategory or field of mathematical signal prepare,

*Corresponding author: shreyasmurthy74@gmail.com

digital concept alter has many benefits over parallel image handle. It admits a wider range of algorithms to be used to the recommendation data and can avoid questions to a degree the addition of crash and distortion all the while treat. Since concepts are defined over two ranges (possibly more) mathematical concept processing can be displayed in the form of intricate systems. An approval method is exploited in which the additives captured as recommendation, is analyzed by way of dataset and the appropriate dishes is urged to the consumer.

2. Content

A. Data Analysis

Data analysis method that focuses on statistical displaying and information discovery for predicting alternatively purely explanatory purposes, while trade intelligence covers data reasoning that relies heavily on collection, attracting mainly on trade facts. In statistical requests, data analysis maybe detached into descriptive enumerations, preliminary data study, and secondary data study. EDA focuses on finding new features in the data while CDA focuses on reinforcing or falsifying existing theories. Predictive science of logical analysis focuses on application of mathematical models for predicting forecasting or categorization, while content analytics applies mathematical, semantic, and structural methods to extract and categorize information from textual beginnings, a variety of unstructured data. All of duplicate are varieties of data reasoning.

B. Feature Extraction

Feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. Feature extraction is related to dimensionality reduction. When the input data to an algorithm is too large to be processed and it is suspected to be redundant (e.g., the same measurement in both feet and meters, or the repetitiveness of images presented as pixels), then it can be transformed into a reduced set of features (also named a feature vector). Determining a subset of the initial features is called feature selection.[1] The selected features are expected to contain the relevant information from the input data, so that the desired task can be performed by using this reduced representation instead of the complete initial data.

C. Training and testing the model

This step includes training and testing of input data. The loaded data is divided into two sets, such as training data and test data, with a division ratio of 80% or 20%, such as 0.8 or 0.2. In a learning set, a classifier is used to form the available input data. In this step, create the classifier's support data and preconceptions to approximate and classify the function. During the test phase, the data is tested. The final data is formed during pre-processing and is processed by the system module.

D. Algorithms Used

This project contains 2 sections.

1) Input data in the text format

Prediction of food recipes by providing ingredients as textual input These applications deal with huge amount of text to perform classification or translation and involves a lot of work on the back end. Transforming text into something an algorithm can digest is a complicated process. In this article, we will discuss the steps involved in text processing. Tokenization converts sentences to words, removing unnecessary punctuation, tags Removing stop words — frequent words such “as”, “the”, “is”, etc. that do not have specific semantic Stemming — words are reduced to a root by removing inflection through dropping unnecessary characters, usually a suffix. Lemmatization — Another approach to remove inflection by determining the part of speech and utilizing detailed database of the language. In text processing, words of the text represent discrete, categorical features Classical ML approaches like ‘Naive Bayes’ or ‘Support Vector Machines’ for spam filtering has been widely used. Deep learning techniques are giving better results for NLP problems like sentiment analysis and language translation. Deep learning models are very slow to train and it has been seen that for simple text classification problems classical ML approaches as well give similar results with quicker training time. The mapping from textual data to real valued vectors is called feature extraction. One of the simplest techniques to numerically represent text is Bag of Words. Classical ML approaches like ‘Naive Bayes’ or ‘Support Vector Machines’ for spam filtering has been widely used. Deep learning techniques are giving better results for NLP problems like sentiment analysis and language translation. Deep learning models are very slow to train and it has been seen that for simple text classification problems classical ML approaches as well give similar results with quicker training time.

a. Recipe Dataset

- We will create our own dataset.
- The dataset will consist of the dish name along with a list of its ingredients, food image and website link.
- Caloric and other basic nutrition content is also added into the dataset.
- Websites considered:
  https://www.indianhealthyrecipes.com
  https://www.indianfoodforever.com
  https://www.archanaskitchen.com

b. K-Nearest Neighbors algorithm

The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. Pause! Let us unpack that. A supervised machine learning algorithm (as opposed to an unsupervised machine learning algorithm) is one that relies on labelled input data to learn a function that produces an appropriate output when given new unlabeled data. Imagine a computer is a child, we are its supervisor (e.g., parent, guardian, or teacher), and we want the child (computer) to learn what a pig looks like. We will show the child several different pictures, some of which are pigs and the rest could be pictures of anything (cats, dogs, etc.). When we see a pig, we shout “pig!” When it’s not a pig, we shout “no, not pig!" After doing this several times with the child, we show them a picture and ask "pig?" and they will correctly (most of the time) say "pig!" or "no, not pig!" depending on what the picture is. That is supervised machine learning. Supervised machine learning algorithms are used to solve classification or regression problems. A classification problem has a discrete value as its output. For example, "likes pineapple on pizza" and "does not like pineapple on pizza" are discrete. There is no middle ground. The analogy above of teaching a child to identify a pig is another example of a classification problem. It is standard practice to represent the output (label) of a classification algorithm as an integer number such as 1, -1, or 0. In this instance, these numbers are purely representational. Mathematical operations should not be performed on them because doing so would be meaningless. The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.

The KNN Algorithm Steps:

1. Load the data
2. Initialize K to your chosen number of neighbors
3. For each example in the data
   3.1 Calculate the distance between the query example and the current example from the data.
   3.2 Add the distance and the index of the example to an ordered collection
4. Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
5. Pick the first K entries from the sorted collection
6. Get the labels of the selected K entries
7. If regression, return the mean of the K labels
8. If classification, return the mode of the K labels.

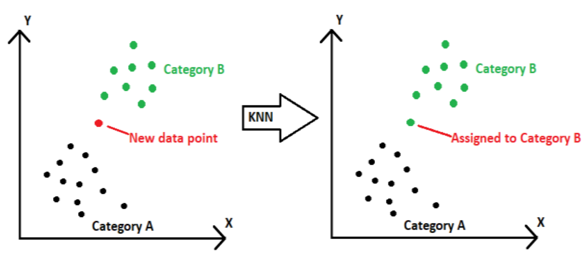

Fig. 1. KNN classification

Choosing the right value for K:

To select the K that's right for your data, we run the KNN algorithm several times with different values of K and choose the K that reduces the number of errors we encounter while maintaining the algorithm's ability to accurately make predictions when it's given data it hasn't seen before. Here are some things to keep in mind: As we decrease the value of K to 1, our predictions become less stable. Just think for a minute, imagine K=1 and we have a query point surrounded by several reds and one green (I'm thinking about the top left corner of the colored plot above), but the green is the single nearest neighbor. Reasonably, we would think the query point is most likely red, but because K=1, KNN incorrectly predicts that the query point is green.

1. Inversely, as we increase the value of K, our predictions become more stable due to majority voting / averaging, and thus, more likely to make more accurate predictions (up to a certain point). Eventually, we begin to witness an increasing number of errors. It is at this point we know we have pushed the value of K too far.
2. In cases where we are taking a majority vote (e.g., picking the mode in a classification problem) among labels, we usually make K an odd number to have a tiebreaker.

Advantages:

- The algorithm is simple and easy to implement.
- There's no need to build a model, tune several parameters, or make additional assumptions.
- The algorithm is versatile. It can be used for classification, regression, and search (as we will see in the next section).

Disadvantages:

- The algorithm gets significantly slower as the number of examples and/or predictors/independent variables increase.

The k-nearest neighbors (KNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. It's easy to implement and understand, but has a major drawback of becoming significantly slows as the size of that data in use grows. 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.

2) Input data as image

Computer vision, the field concerning machines being able to understand images and videos, is one of the hottest topics in the tech industry. Robotics, self-driving cars, and facial recognition all rely on computer vision to work. At the core of computer vision is image recognition, the task of recognizing what an image represents. Before performing any task related to images, it is almost always necessary to first process the images to make them more suitable as input data. In this article I will focus on image processing, specifically how we can convert images from JPEG or PNG files to usable data for our neural networks. Then, in other articles I will concentrate on the implementation of classic Convolutional Neural Network or some specific ones as ResNet and SqueezeNet. The library we will use is TensorFlow 2.0 as it provides a variety of utility functions to obtain image data from files, resize the images, and even transform a large set of images all at once. Image file Before we do any image processing, we need to understand how image files work. Specifically, we'll discuss how these files use byte data and pixels to represent images. If you've ever looked at an image file's properties before, it'll show the dimensions of the image, i.e., the height and width of the image. The height and width are based on number of pixels. For example, if the dimensions of an image are 400x300 (width x height), then the total number of pixels in the image is 120000.The soccer ball image has dimensions 1710x1980 (1710px width, 1980px height), while the tennis ball image is 1024x1024 (1024px width, 1024px height). The function tensorflow.io.read_file takes the file name as its required argument and returns the contents of the file as a tensor with type tensorflow.string. When the input file is an image, the output of tensorflow.io.read_file will be the raw byte data of the image file. Although the raw byte output represents the image's pixel data, it cannot be used directly. Let's first see the implementation in Python using the soccer ball image. Dataset Normally when we do image related tasks, we're dealing with a large amount of image data. In this case, it's best to use a TensorFlow dataset, i.e., tensorflow.data.Dataset, to store all the images. We can create a dataset using the from_tensor_slices function. The Dataset class makes it easier and more efficient to perform tasks with all the image files. After we create a dataset with the image files, we will need to decode each file's contents into usable pixel data. Since the decode_image function works for single image files, we will need to use the dataset object's map function to apply decode. image to each image file in our dataset.

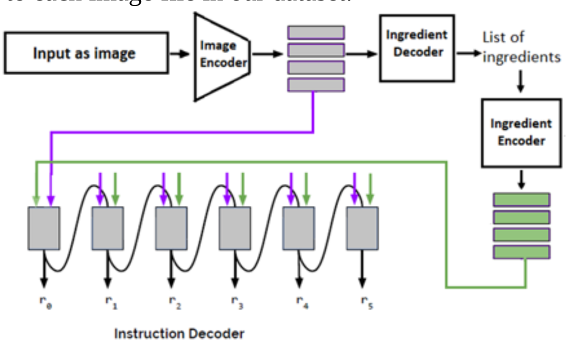

Fig. 2.

E. Keras

Keras is a minimalist Python library for deep learning that can run on top of Theano or TensorFlow. It was developed to make implementing deep learning models as fast and easy as possible for research and development. It runs on Python 2.7 or 3.5 and can seamlessly execute on GPUs and CPUs given the underlying frameworks. It is released under the permissive MIT license. Keras was developed and maintained by François Chollet, a Google engineer using four guiding principles:

- Modularity: A model can be understood as a sequence or a graph alone. All the concerns of a deep learning model are discrete components that can be combined in arbitrary ways.
- Minimalism: The library provides just enough to achieve an outcome, no frills and maximizing readability.
- Extensibility: New components are intentionally easy to add and use within the framework, intended for researchers to trial and explore new ideas.
- Python: No separate model files with custom file formats. Everything is native Python.

Keras proper does not do its own low-level operations, such as tensor products and convolutions; it relies on a back-end engine for that. Even though Keras supports multiple back-end engines, its primary (and default) back end is TensorFlow, and its primary supporter is Google. The Keras API comes packaged in TensorFlow as tf.keras, which as mentioned earlier will become the primary TensorFlow API as of TensorFlow 2.0. The Model is the core Keras data structure. There are two main types of models available in Keras: The Sequential model, and the Model class used with the functional API. Keras Sequential models The Sequential model is a linear stack of layers, and the layers can be described very simply. Here is an example from the Keras documentation that uses model.add() to define two dense layers in a Sequential model

F. Data Pre-Processing

Data Pre-processing is a method that is used to convert the raw data into a clean basic document file. In other words, at any time the data is assembled from various beginnings it is collected in inexperienced layout that is not possible for the analysis. In the final data frame, skilled are two categorical lines in the data frame, categorical data are variables that contain label principles alternatively mathematical values. The number of attainable principles is frequently limited to an established set, like in this place case, parts and nations values. Many machine intelligence algorithms cannot function on label data directly, they demand all recommendation variables and manufacturing variables expected numeric. This wealth that unconditional data must be converted to a mathematical form. The unconditional profit shows the numerical worth of the effort in the dataset. This encrypting will create a twofold procession each classification and returns a matrix accompanying the results

G. Ingredient Classification

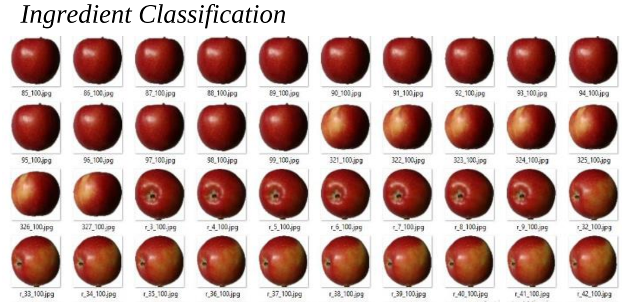

Fig. 3. Apple image dataset to train the data

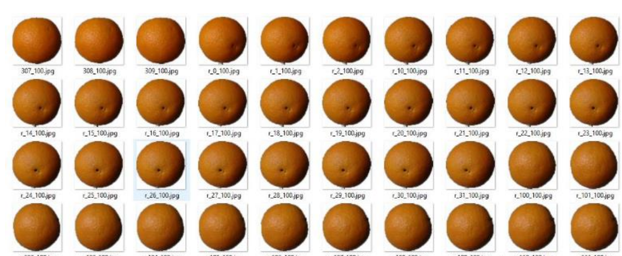

Fig. 4. Orange image dataset to train the data

Training Data In machine learning, a common task is the study and construction of algorithms that can learn from and make predictions on data. Such algorithms function by making data-driven predictions or decisions, through building a mathematical model from input data. The model is initially fit on a training dataset, which is a set of examples used to fit the parameters of the model. The model is trained on the training dataset using a supervised learning method, for example using optimization methods such as gradient descent or stochastic gradient descent. In practice, the training dataset often consists of pairs of an input vector (or scalar) and the corresponding output vector (or scalar), where the answer key is commonly denoted as the target. The current model is run with the training dataset and produces a result, which is then compared with the target, for each input vector in the training dataset. Based on the result of the comparison and the specific learning algorithm being used, the parameters of the model are adjusted. The model fitting can include both variable selection and parameter estimation. Successively, the fitted model is used to predict the responses for the observations in a second dataset called the validation dataset. The validation dataset provides an unbiased evaluation of a model fit on the training dataset while tuning the model's hyper parameters.

H. Testing Data

The test set is a set of observations used to evaluate the performance of the model using some performance metric. It is important that no observations from the training set are included in the test set. A test dataset is a dataset that is independent of the training dataset, but that follows the same probability distribution as the training dataset. If a model fit to the training dataset also fits the test dataset well, minimal over fitting has taken place.

I. Modeling

1) K-Nearest Neighbor

KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry. To evaluate any technique, we generally look at 3 important aspects:

1. Ease to interpret output
2. Calculation time
3. Predictive Power Simple case to understand this algorithm.

Following is a spread of red circles (RC) and green squares intend to find out the class of the blue star.

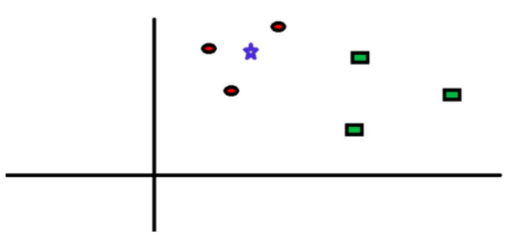

Fig. 5. KNN classification representation

BS can either be RC or GS and nothing else. The "K" is KNN algorithm is the nearest neighbor we wish to take the vote from. Let's say K = 3. Hence, we will now make a circle with BS as the center just as big as to enclose only three datapoints on the plane. Refer to the following diagram for more details.

The three closest points to BS are all RC. Hence, with a good confidence level, we can say that the BS should belong to the class RC. Here, the choice became very obvious as all three votes from the closest neighbor went to RC. The choice of the parameter K is very crucial in this algorithm. Next, we will understand what are the factors to be considered to conclude the best K. last example, given that all the 6-training observation remain constant, with a given K value we can make boundaries of each class. These boundaries will segregate RC from GS. In the same way, let's try to see the effect of value "K" on the class boundaries. The following are the different boundaries separating the two classes with different values of K.

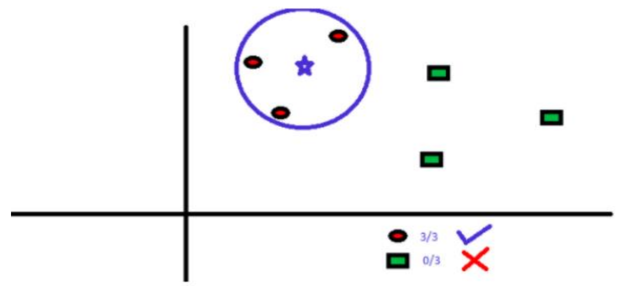

Fig. 6. Representing the BS among cluster

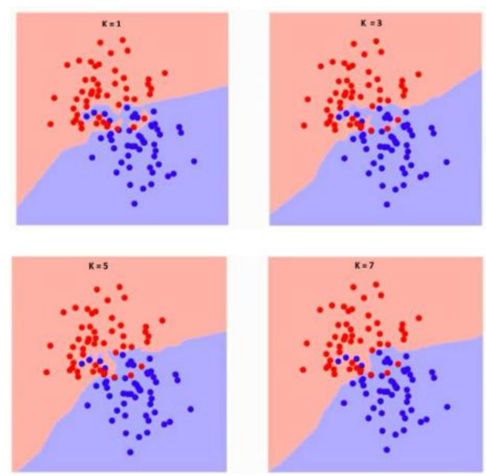

Fig. 7. Boundaries separating the two classes

The boundary becomes smoother with increasing value of K. With K increasing to infinity, it finally becomes all blue or all red depending on the total majority. The training error rate and the validation error rate are two parameters we need to access different K-value. Following is the curve for the training error rate with a varying value of K.

The error rate at K=1 is always zero for the training sample. This is because the closest point to any training data point is itself. Hence the prediction is always accurate with K=1. If validation error curve would have been similar, our choice of K would have been 1. Following is the validation error curve with varying value of K.

This makes the story clearer. At K=1, we were overfitting the boundaries. Hence, error rate initially decreases and reaches a minima. After the minima point, it then increases with increasing K. To get the optimal value of K, you can segregate the training and validation from the initial dataset. Now plot the validation error curve to get the optimal value of K. This value of K should be used for all predictions.

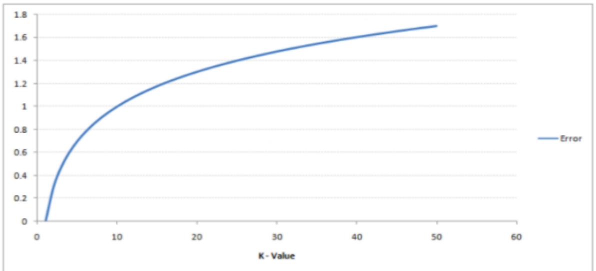

Fig. 8. Training error rate curve

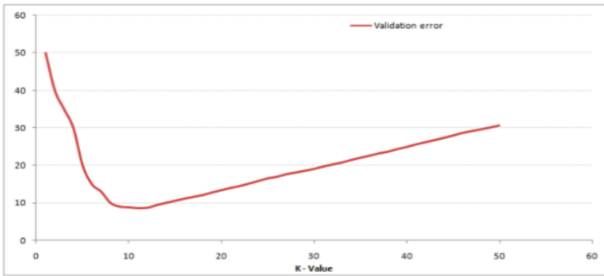

Fig. 9. Validation error curve

3. Results

A Django webapp is developed, a fully featured Python web framework that can be used to build complex web applications. Food Recipe detection have gained a great deal of popularity due to the wide range of applications that they have proved to be useful in. Broadly, two main categories for these applications exist: commercial applications and research applications.

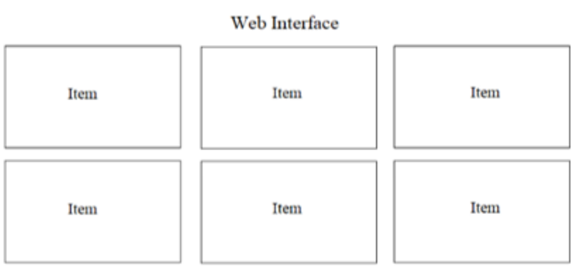

Fig. 10. Web interface

The list of all the recipes which can be prepared easily using the available ingredients will be shown at the end of the result. User can be able to rate the recipe according to their wish and they can even upload the image of the ingredients available and able to predict the recipe.

4. Conclusion

The LeNet construction favorably recognizes the various figure recommendation accompanying a very high veracity. Our KNN classifier classifiers the recipes efficiently. As future enhancements we would like to extend our image dataset concerning recognition of more elements.

We would still like to decide recipes establishing various cuisines. From the literature, we find the evidence that plurality the culture of the realm is unhealthy. In this paper various datasets miscellaneous concept classification indicates and processes the methods of KNN.

Though the project has a lot of potential on its own, the significance lies in the multitude of diverse applications that can be processed through it. One of the key applications is enhancement of physical appearance of food. Food is the key element of economical aspect in several fields like Hotel Management, which can be ensured through an algorithm that can easily determine ingredients based on their physical attributes like color, shape, etc. Also having a method to determine the least cost ingredients required to cook a certain recipe can result in a massive increase in profit for businesses depending on food that are still in their growing stages. As an additional benefit, such algorithms easily ensure the preparation is at its utmost hygienic conditions, since we are already aware of how sanitization is the pre-requisite of safety considering recent circumstances with COVID-19.

With an exceptional algorithm, however, we still need the human touch to ensure the dishes prepared can attain perfection. The input of the cook to be able to taste and reconfigure the recipe accordingly and set the algorithm on re-analysis would be a huge contribution as of future aspect concerned. If such a combination of human input and algorithm is achieved, the efficiency is enhanced to a great extent.

Using Classification dataset, an arrangement is grown where several sources of data are likely treated as input and a direction that uses only those ingredients is urged. This structure picks most efficient directions in accurate manner. With the help of machine learning, the finding of new formula using feasible elements by the consumer may be used in a more efficient manner.

References

[1] Amaia Salvador, Michal Drozdzal, Xavier Giro-i-Nieto and Adriana Romero, "Inverse Cooking: Recipe Generation from Food Images", *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

[2] Wei-Ta Chu and Jia-Hsing Lin, "Food Image Description Based on Deep-Based Joint Food Category, Ingredient, And Cooking Method Recognition", *Proc of the IEEE International Conference on Multimedia and Expo Workshops (ICMEW),* 2017.

[3] A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Ofli, I. Weber, and A. Torralba, "Learning cross-modal embeddings for cooking recipes and food images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2017.

[4] Punnarumol Temdee, Surapong Uttama," Food Recognition on Smartphone using Transfer Learning of Convolution Neural Network", *IEEE,* 2017.

[5] Raza Yunus, Omar Arif, Hammad Afzal, Muhammad Faisal Amjad," A Framework to Estimate the Nutritional Value of Food in Real Time Using Deep Learning Techniques", *IEEE Access,* 2018.

[6] Prof. M.P Nerkar, Preety Vajpayee, Varsha Ganjoo, Vaibhav Suryawanshi, Pooja Sonawane, "Recipe Retrieval Using Ingredient Recognition", *IJAERD,* Volume 4, Special Issue 5, Dec. 2017.

[7] Mohammed A. Subhi, Sawal Md. Ali, "A Deep Convolutional Neural Network for Food Detection and Recognition", *IEEE-EMBS Conference on Biomedical Engineering and Sciences,* 2018.

[8] Sanasam Inunganbi, Ayan Seal, and Pritee Khanna, "Classification of Food Images through Interactive Image Segmentation", *Springer International Publishing AG, part of Springer Nature,* 2018.

[9] W. Min, B. K. Bao, S. Mei, Y. Zhu, Y. Rui, and S. Jiang, "You are what you eat: Exploring rich recipe information for cross-region food analysis," *IEEE Transactions on Multimedia,* vol. PP, no. 99, p. 1, 2017.

[10] J. Chen, L. Pang, and C.-W. Ngo, "Cross-modal recipe retrieval: How to cook this dish?" in *International Conference on Multimedia Modeling. Springer,* 2017, pp. 588–600.