JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY

FOOD RECOGNITION AND CALORIE ESTIMATION

REPORT

B2 BATCH

Submitted by:
ITI JAIN(15103143)
SAMYAK JAIN(15103154)
UDAY AGGARWAL(15103163)

Submitted to:
MS. MEGHA RATHI

4th May 2018

TABLE OF CONTENTS

ABSTRACT

In this project , we made a food recognition and calorie estimation system that uses the images of the food, given by the user, to recognize food item and then estimates the calorie present in same food item. Food image recognition is one of the promising applications of visual object recognition in computer vision.
The system uses image processing and computational intelligence for food item recognition.
We trained a large, deep convolutional neural network to classify the 1000 high-resolution images of each category. The classifiers for SVM, Random Forest and KNN had already been trained prior to the deep neural networks and the performance of each Machine Learning Algorithm in terms of speed and accuracy are compared to determine the best possible classifier for our dataset.

INTRODUCTION

Well-being is becoming a topic of great interest and an essential factor linked to improvements in the quality of life. Modern information technologies have brought a new dimension to this topic. Due to the widespread use of low-cost imaging devices like smartphone cameras, more and more applications are being developed in computer vision to facilitate automatic object recognition, among which food image recognition has recently gained much attention. Nowadays, people, especially diabetes patients, are increasingly cautious about their diet for improved health care. Food image recognition provides a simple means to estimate the dietary caloric intake and evaluate people's eating habits, by using cameras to keep track of their food consumption. An accurate estimation of daily nutritional intake provides a useful solution for keeping healthy and to prevent diseases.

CHALLENGES

In the last couple of years,advancements in image processing, machine learning and in particular deep learning, and convolutional neural network (CNN) proved to be a boon for the image classification and recognition tasks, including for the problem of food image recognition. Researchers have been working on different aspects of a food recognition system, but there is still a lack of good-enough solution to high-accuracy food classification and recognition.Therefore, it is extremely difficult to correctly recognize every food item,as many of the food items may look similar in color or shape and are not even distinguishable to human eyes.Therefore, we state that it would be good enough to recognize the general type of a certain food item, based on which we can approximately estimate its dietary value, e.g., calories. It can already provide people with basic information on their daily intake.

Also, as the project scales more and more, the specific food items can also be encapsulated with specific or detailed information which were previously generalized.

METHODOLOGY

The project consist of two steps:identifying food from an image,and converting the food identified into a calorie estimation.We performed food image classification using CNN(Convolutional Neural Network).

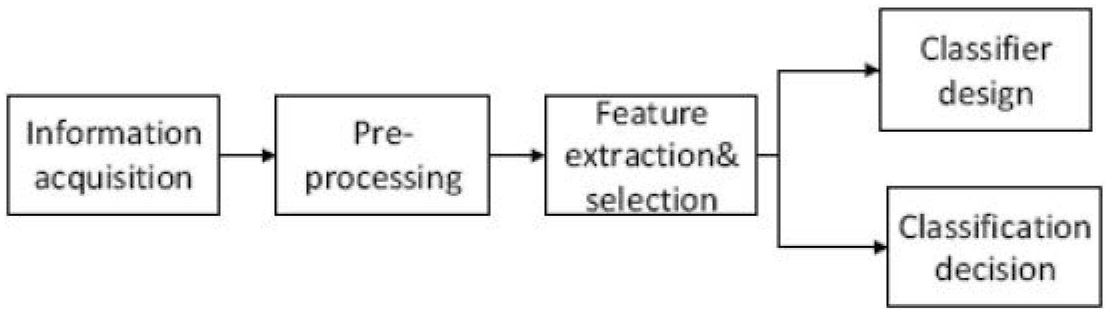
Steps followed:

- **Preprocessing :** Some basic preprocessing has been performed to clean the dataset where the irrelevant and noisy images of 15 categories have been removed. Also, data augmentation has been performed -
  - Pixel values rescaled in the range of [0,1].
  - Random rotations upto 40 degree.
  - Random zoom applied.
  - Shear angle in counter-clockwise direction in degrees

- **Trained the model :** We trained the model of images for 15 categories using the classifier CNN(Convolutional Neural Network) which is a class of deep, feedforward artificial neural networks that has successfully been applied to analyzing visual imagery.

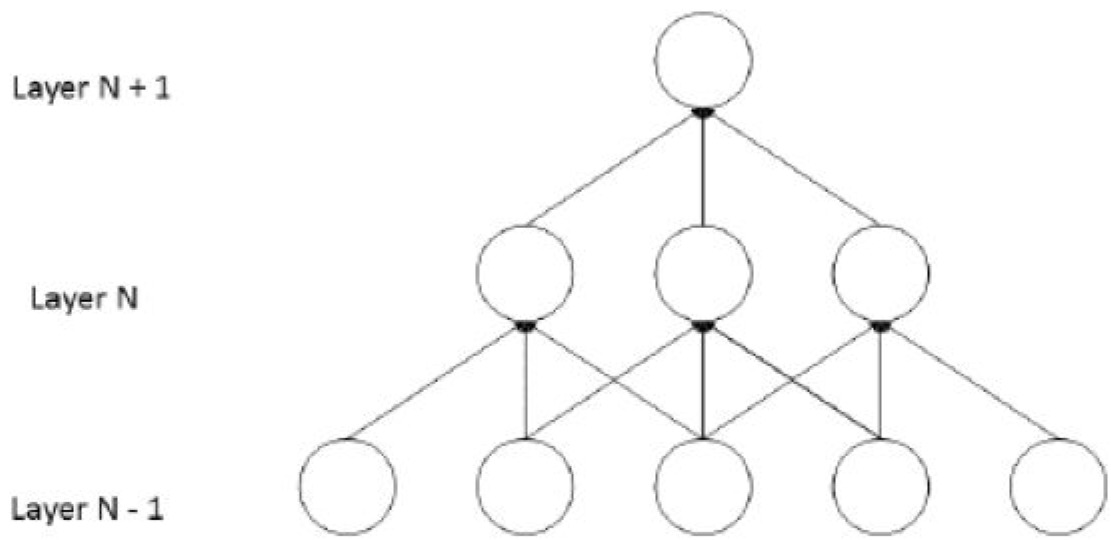
CONVOLUTIONAL NEURAL NETWORK

The Convolutional Neural Network (CNN) offers a technique for many general image classification problems. It has been applied in food classification and resulted in a high accuracy.CNN is widely used in food recognition and provides better performance than the conventional methods.



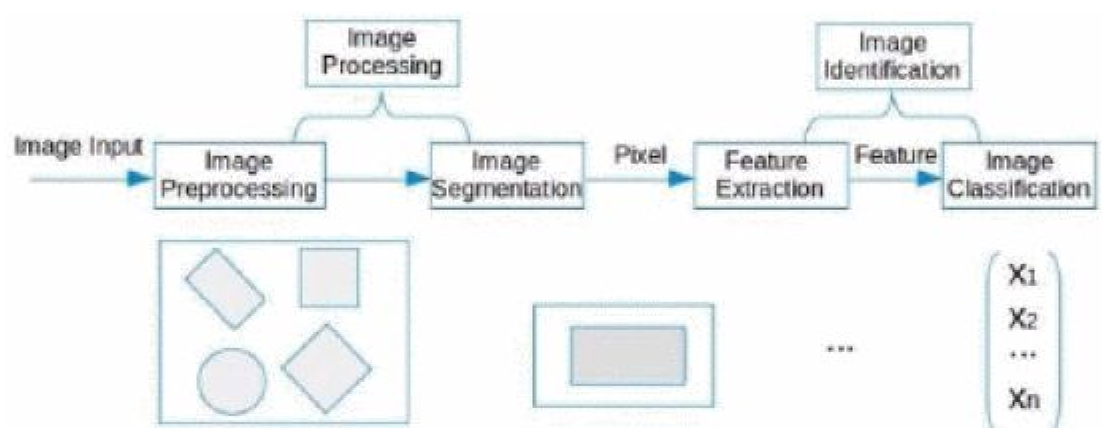
Tradition Model and CNN

Over the last few years, due to the advancements in the deep learning, especially in the convolutional neural networks, the accuracy in identifying and recognizing images has been increased drastically. This is not only because larger datasets but also new algorithms and improved deep architectures. Convolutional Neural Network (CNN) is also known as LeNet due to its inventor.CNN mainly comprises convolutional layers, pooling layers and sub-sampling layers followed by fully-connected layers. The very first architecture of CNN takes an input image and applies convolution followed by sub-sampling. After two such computations, the data is fed into the fully connected neural network,where it performs the classification task. The main advantage of CNN is the ability to learn the high-level efficient features and in addition to that, it is robust against small rotations and shifts.



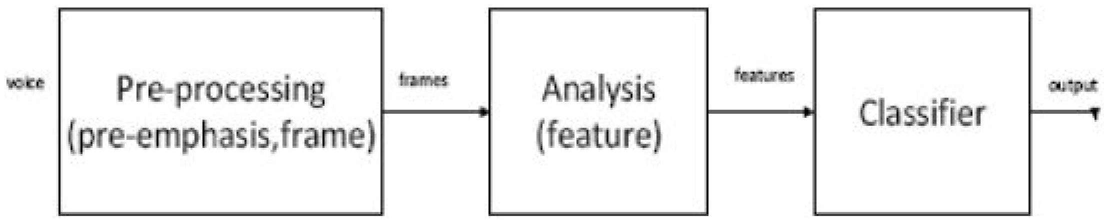
The schematic diagram of sparse connectivity

In pattern recognition perspective image recognition can be abstracted into the form in below figure.



Four steps of image recognition

Specific to the field of image recognition the process can be summarized in below figure.



FOOD 101 DATASET

The dataset contains a number of different subsets of the full food-101 data. For this reason the data includes massively downscaled versions of the images to enable quick tests. The data has been reformatted as HDF5 and specifically Keras HDF5Matrix which allows them to be easily read in. The file names indicate the contents of the file.There are 101 categories represented, with 1000 images, and most have a resolution of around 512x512x3 (RGB, uint8).

We have used 15 categories in our project. They are Apple Pie, Club Sandwich, Grilled Cheese Sandwich, Tacos, Hamburger, Samosa, French Fries, Pizza, Ravioli, Cake, Spring Rolls,

PACKAGES USED

(i) KERAS

Keras is a high-level neural networks API developed with a focus on enabling fast experimentation. *Being able to go from idea to result with the least possible delay is key to doing good research.* Keras has the following key features:

- Allows the same code to run on CPU or on GPU, seamlessly.
- User-friendly API which makes it easy to quickly prototype deep learning models.
- Supports arbitrary network architectures: multi-input or multi-output models, layer sharing, model sharing, etc. This means that Keras is appropriate for building essentially any deep learning model, from a memory network to a neural Turing machine.
- Is capable of running on top of multiple back-ends including Tensorflow, CNTK, or Theano.

(ii) JPEG

This package provides an easy and simple way to read, write and display bitmap images stored in the JPEG format. It can read and write both files and in-memory raw vectors.

(iii) CARET

The Caret package (short for _C_lassification _A_nd _RE_gression _T_raining) is a set of functions that attempt to streamline the process for creating predictive models. The package contains tools for:

- data splitting
- pre-processing
- feature selection
- model tuning using resampling
- variable importance estimation

(iv) SHINY

Shiny is an R package that makes it easy to build interactive web apps straight from R. This package can host standalone apps on a webpage or embed them in R Markdown documents or build dashboards This package can also extend Shiny apps with CSS themes, htmlwidgets, and JavaScript actions.

(v) SHINYJS

Perform common useful JavaScript operations in Shiny apps that will greatly improve your apps without having to know any JavaScript. Examples include: hiding an element, disabling an input, resetting an input back to its original value, delaying code execution by a few seconds, and many more useful functions for both the end user and the developer. 'shinyjs' can also be used to easily call your own custom JavaScript functions from R.

(vi) SHINYTHEMES

Bootstrap can use alternative CSS files in place of the stock bootstrap.css file. When the shinythemes package is loaded, it makes these alternate themes available to Shiny applications in a relative URL under shinythemes.

(vii) DEVTOOLS

Devtools makes package development a breeze: it works with R’s existing conventions for code structure, adding efficient tools to support the cycle of package development. With devtools, developing a package becomes so easy that it will be your default layout whenever you’re writing a significant amount of code.

(viii) RSCONNECT

The rsconnect package provides a programmatic deployment interface for RPubs, shinyapps.io, and RStudio Connect. Supported contents types include R Markdown documents, Shiny applications, plots, and static web content.

(ix) CATOOLS

Contains several basic utility functions including: moving (rolling, running) window statistic functions, read/write for GIF and ENVI binary files, fast calculation of AUC, LogitBoost classifier, base64 encoder/decoder, round-off-error-free sum and cumsum, etc.

(x) E1071

Functions for latent class analysis, short time Fourier transform, fuzzy clustering, support vector machines, shortest path computation, bagged clustering, naive Bayes classifier.

(xi) KNN

This function provides a formula interface to the existing knn() function of package class. On top of this type of convenient interface, the function also allows normalization of the given data.

MODEL ARCHITECTURE

The first layer is the Convolutional 2D layer which consists of 32 kernels of size 3x3 taking an input of size 100x100x3 where 100x100 is the rescaled size of our images and 3 denotes the color aspect (RGB) of the image.

The next layer is the max pooling layer with a pool size of 2x2.

The above two layers are again repeated to get better filtered convolved images and better feature extraction by the max pooling layer.

The above layers have been repeated one last time where the kernels have been increased from 32 to 64 to get more filtered images for the fully connected layers.

Two fully connected layers are used next with 128 and 90 neurons respectively and Dropouts have been added of 0.01 in between the dense layers to prevent overfitting by making the weights of some random neurons to zero so as to prevent overfitting on some particular neurons.

All the convolutional 2D layers and the fully connected layers have an activation function of ReLu (Rectified Linear Unit).

The last layer is the output layer consisting of 15 neurons equivalent to the number of categories and each neuron has an output of a probability corresponding to that particular neuron. The CNN predicts the category to be the one with the highest probability.

RESULT

By the application of various machine learning algorithms (Support Vector Machines, K-Nearest Neighbour, Random Forest Classification) and deep learning algorithm(Convolutional Neural Networks) on the image classification problem, it is concluded that CNN is the most viable method for the image classification on our dataset in terms of both speed and accuracy and that CNN performs best in the cases of large datasets.

CONCLUSION

The traditional models in machine learning are weaker in practice when it comes to image classification. In this project, the CNN model is applied in image recognition and comparison was made between different classic algorithms and CNN. A lot of data augmentation and segmentation has to be performed as well clean pixel values which is not necessary in CNN as it on it's own learn the generalized pattern required to identify and recognize unseen images. Shiny package was used to build simplistic yet powerful UI.

FUTURE WORK SCOPE

(i) More categories can be added into the model and make the classifier predict on a wider variety of different food items.
(ii) Can recognize multiple count food items in the same meal, such as hamburgers and cakes on the same dish, to estimate the calories of the meal in a more efficient manner.
(iii) Can make the platform more custom based for different users, by learning from different users their preferred food categories and making suggestions for different restaurants or any eating place for the same.

REFERENCES

I. https://www.kaggle.com/kmader/food41/data
II. https://www.researchgate.net/publication/310823982_FoodNon-food_Image_Classification_and_Food_Categorization_using_Pre-Trained_GoogLeNet_Model
III. https://www.researchgate.net/publication/319070943_Mobile_Multi-Food_Recognition_Using_Deep_Learning
IV. https://www.researchgate.net/publication/319770183_Imagenet_classification_with_deep_convolutional_neural_networks
V. https://keras.rstudio.com
VI. https://ieeexplore.ieee.org/document/7550778/
VII. https://www.vision.ee.ethz.ch/datasets_extra/food-101/
VIII. http://visiir.univ-lr.fr/images/publication/CEA_ICME2015.pdf
IX. https://www.jstatsoft.org/article/view/v015i09
X. https://arxiv.org/ftp/arxiv/papers/0802/0802.2411.pdf
XI. https://ieeexplore.ieee.org/document/6123373/?reload=true
XII. https://medium.com/@Synced/deep-learning-based-food-calorie-estimation-method-in-dietary-assessment-1e76a2acee7
XIII. https://shiny.rstudio.com
XIV. https://www.researchgate.net/publication/311370165_Food_Image_Recognition_by_Using_Convolutional_Neural_Networks_CNNs