CALORIE COUNTER USING IMAGE RECOGNITION

Pratyush Sachan[1], Rachit Kumar[2], Pranjul Pandey[3], Prawal Gupta[4], Dr. Suneetha K R[5]
Department of Computer Science & Engineering,
Bangalore Institute of Technology , K.R. Road , V.V. Puram ,
Bengaluru-560004,
Email - principalbit4@gmail.com
[1]spratyush.1998@gmail.com, rachit016@gmail.com, pranj99@gmail.com, prawalgupta171@gmail.com, suneetha.bit@gmail.com

Abstract. Image Processing in the recent decades has been revolutionized by the development in the domain of Convolutional Neural Network. In 2012 when Alex Krizhevsky won the ILSVRC 2012 with his AlexNet model, it marked a beginning of a phase where in the coming 4-5 years huge improvements were made in the CNN architectures. Leveraging Neural Networks and Computer Vision in the health sector has seen a huge increase in trend in the recent past with many web and mobile applications being developed for diet management or food recognition etc. In this work, we have implemented a android based app that will classify the image of food that we input and then display the associated calories The app was developed using Flutter Framework with underlying CNN base architecture as ResNet50 configured with some manual interventions like addition of dropout layers etc trained over Indian Food Image Dataset that we scrapped manually. The model performed reaaly well with traing accuracy of 88% and validation accuracy of 79% and was successfully classifying the food images and displaying their calories.

Keywords: Image Processing, Convolutional Neural Network, Deep Learning, ResNet, Flutter, Food

1. Introduction

Food in today's scenario, rather than being just a thing to eat, has an aesthetic presence around us. The commodity of food has evolved in taste, texture. Today focus is more on the presentation side of the food. So it's our responsibility to know what we are eating and how it affects our body. Dietary assessment has been in focus for the past one decade and we are today more conscious about our eating habits. In the past, the assessment was done majorly using oral means where the patient was monitored for a day or two, and was asked to tell the food items that they took for those days. The major limitation of this procedure was that it suffered from a certain bias because the result was solely dependent on the retention power of the subject for what he/she has eaten for proposed period of time. Then came the machine learning approach which was far more impactful than the previous approach but again the feature detection part and the fine tuning was done manually which again was a hefty task. Now is the era of Deep learning.

Today the domain of Computer vision has taken a giant leap and so has Deep neural networks. Leveraging deep learning techniques in the computer vision domain has worked wonders in the past. Deep learning models have made great strides in variety of computer vision problems such as Object Detection, Face Recognition, Object classification and has outperformed state of art machine learning techniques in this field. Also, the empowerment of parallel GPU computing over cloud has made training the neural networks much easier and faster than it was before.

In this paper, we discuss the system for automatically tracking the calorie of a particular food item by clicking its picture or uploading it through the gallery. The process of staying healthy is now much easier and a person can monitor the calorie of the food he/she is consuming in a snap. The system will be a mobile based application in which we will be using Convolutional Neural Networks as our

deep learning model which has been hugely successful in computer vision applications. To be precise we will be using ResNet50 as our underlying CNN architecture. The novel part of our system would be the use of Flutter as our application development framework which is faster and more scalable than other frameworks like React Native, Xamarin etc.

2. Related Work

[1]Chang Liu et al., 2014 have proposed a CNN model using supervised learning algorithms. The structure was inspired by LeNet, GoogleNet models. They have fed 32x32 grayscale image as input to the network and then applied several convolutional and sampling layers and in the end using 2 fully connected layers generated a 10 class output. Also addition of a 1x1 convolutional layer helped in reduced bottleneck and capturing more visual information.

[2]Siddarth Sairaj et al., 2020 proposed a CNN model using ResNet architecture to overcome drawbacks of VGG model where generalisation becomes a problem whereas ResNets perform classification optimally. After that a React Native rendering framework displays the information about the food item on the screen.

[3]Alex Krizehevsky et al., 2012 proposed and developed a new CNN model known as AlexNet in the ImageNet challenge. The model had a significant performance improvement as compared to other models. Their top-5 error rate was 15.8% which was far better compared to the next best of 26.2%. The overfitting issue which was their in the training phase was eliminated using Data augmentation and Dropout layer.

[4]L Jiang et al., 2020 proposed a system for using Faster R-CNN for food identification and detection. He used VGG-16 to extract feature maps from generated region proposals and classify them as food item. Also they used regression to locate food in the image. After detecting the food item, it is analysed and report is summarised based in modern dietary assessment tools.

[5]Kaiming He et al., 2015 proposed one of the first papers that used ResNets extensively in form of the shorter connections to deeper layer. Also it compared ResNets to plain networks. The paper provides comprehensive empirical evidence showing that residual networks are easier to optimise, and can gain accuracy from considerably increased depth.

[6]Andrew Zisserman et al., 2015 investigated the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Their main contribution was a thorough evaluation of networks of increasing depth using an architecture with very small (3 × 3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16–19 weight layers. These findings secured them 1st and 2nd place in ImageNet challenge 2014.

[7]Seon-Joo Park et al., 2019 proposed a model that trained on more than 4000 Korean food images. Their complex food recognition model, K-foodNet, had higher test accuracy (91.3%) and faster recognition time (0.4 ms) than those of the other networks. In the architecture Convolutional-pooling layers using 3 × 3 kernel and producing 256 and 512 feature maps are followed by the max polling layers. After the last pooling layer, the fully connected (FC) layers are activated with 2048 nodes as well as the dropout layer between the FC layers.

[8]Stergios Christodoulidis et al., 2015 proposed a system for the recognition of already segmented food items in meal images using a deep CNN, trained on fixed-size local patches. Our approach exploits the outstanding descriptive ability of a CNN, while the patch-wise model allows the generation of sufficient training samples, provides additional spatial flexibility for the recognition and ignores background pixels.

[9]Xi-Jin Zhang et al., 2016 proposed a system where he built a dataset of 250 000 images of 360 categories of foods. We developed an automatic outlier elimination method employ-ing deep convolutional features. A multi-task DCNN system was proposed and achieved 57.25% in the top-1 accuracy and 82.29% in the top-5 accuracy for the dish identification task. The result outperforms the traditional SVM method significantly.

[10]Abdulkadir Senguri et al., 2019 used deep feature extraction, feature concatenation and support vector machine (SVM) classifier for efficient classification of food images. Classification of foods according to their

images becomes a popular research task for various reasons such as food image retrieval and image based self-dietary assessment. For deep feature extraction, pre-trained AlexNet and VGG16 models are considered. The concatenated features are then classified with SVM. After fine-tuning of the model it observed an accuracy of 79.86%.

[11]Kuang Huei Lee et al., 2018 studied the problem of learning image classification models with label noise. They introduced CleanNet, a joint neural embedding network, which only requires a fraction of the classes being manually verified to provide the knowledge of label noise that can be transferred to other classes and integrate CleanNet and conventional convolutional neural network classifier into one framework for image classification learning.

[12]Ashutosh Singla et al., 2016 reported experiments on food/non-food classification and food recognition using a GoogLeNet model based on deep convolutional neural network. The experiments were conducted on two image datasets created by our own, where the images were collected from existing image datasets, social media, and imaging devices such as smart phone and wearable cameras. Experimental results show a high accuracy of 99.2% on the food/non-food classification and 83.6% on the food category recognition.

3. Dataset Preparation

The database majorly consists of Indian food item labelled into different classes. The images were scrapped from image search engines such as google and photo sharing application such as Pinterest and Instagram. The scrapping was done using a Python web crawler written by us. Some existing open source datasets from Kaggle were also included. We collected images for 71 classes of food item with every class having around 300-400 images. The acquired images have resolutions in the range of a minimum of 200x150 to 5760x3840 pixels per image. The model always works better if we have diversified data and so we have applied Data Augmentation to our dataset which is an approach for generating more synthetic data from the original images using activities like resizing, zoom, padding, rotating and etc. This process of augmentation makes the model robust to deal with the variations in the real world. 70% of the images contributed to the training set, and the remaining 30% was used for the purpose of testing. A cross-validation based approach was followed for the proper use of available data and to improve the accuracy of the model.

4. Hardware Specifications

Image classifier model training(Google Collaboratory)

1. GPU: 1xTesla K80, compute 3.7, having 2496 CUDA cores
2. CPU: Quad-core hyper-threaded 10th gen. Intel Core i5 Processor, 4 core, 2 threads.

Hosting Platform

Google Cloud Platform cloud computing infrastructure.

5. Proposed System

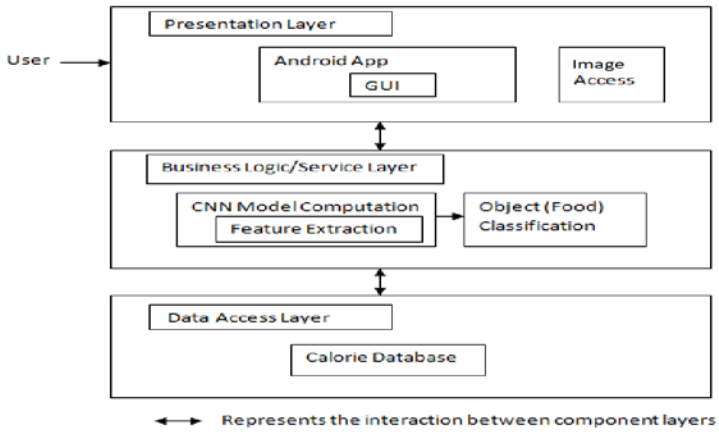

Fig 5.1 Component Diagram of the system

The above fig 5.1 , depicts how the layers in the system interact with each other. Starting off with the presentation layer or the UI which consists of GUI and camera/gallery access features which subsequently interact with underlying business logic layer that consists of the CNN model trained
And which classifies the food item and then to fetch the calories interact with data access layer.

In order to create a classifier for identifying the food images, a deep learning model was used. The performances of a variety of models were compared before finally selecting a model which could classify images correctly with a certain level of confidence.We started by creating a shallow architecture which did not have sufficient parameters to learn. Consequently, this model was underfitting the data by a large extent and we were only able to muster up an accuracy of about 40%.

Then we directly switched over to transfer learning to examine the performances on our dataset. We chose Inception V3 as our base model and the top layer was a custom Softmax layer to make a prediction for the 71 classes of food at our disposal. All the layers in the base model were frozen and a Dense layer with 128 neurons and a Flat layer were added with dropouts of 0.2 and 0.5 respectively which were the trainable layers in our model. The accuracy of this model peaked at about 62%, underfitting the data yet again.
The reason for the model underfitting was that InceptionV3 has been trained on a dataset very much different from the data which our food image dataset was comprised of. Whatever the base model learned was not suitable for our own dataset and the final layers weren't enough to learn enough for the data available.

5.1 Neural Network Architecture

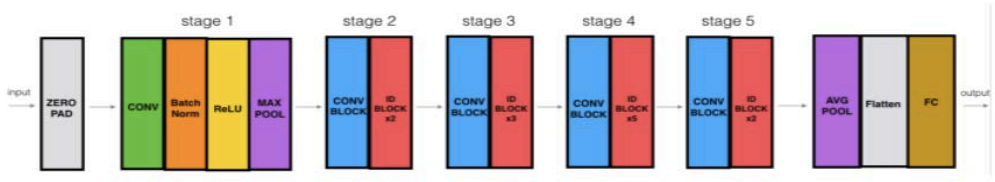

Fig 5.1.1 ResNet Architecture

After the comparisons being made and the conclusion being drawn out from the models examined, we finally decided upon choosing ResNet50 as the base architecture. The entire model was kept trainable which is the opposite of what is done in transfer learning and the whole model was trained from the ground up.
A global average pooling layer was added to the base model with a dropout of 0.2 and a Dense layer with 128 neurons were added to it with a dropout of 0.5. Finally, a softmax layer was added to predict the image on the basis of the 71 classes. An l2 regularizer was added in this layer to penalise the weights helping the model to prevent overfitting along with the dropout layers provided.

A stochastic gradient descent optimizer was used with a relatively low learning rate of 0.0001 and a momentum of 0.9. The loss function used here was categorical crossentropy which is suitable for a dataset with large number of classes. Due to unavailability of a good GPU, this model was trained of Goolge Colab's GPU which keeps disconnecting after a few hours. Consequently, multiple stretches were used to train this model and a ModelCheckpoint was passed so that the model keeps saving itself whenever the validation loss decreased. This helped to start the training from where it was left in the previous run.

The model contains 23,859,143 total parameters, 23,806,023 trainable parameters and 53,120 non trainable parameters.
The training accuracy reached upto a maximum of 88% and the validation accuracy reached a maximum of 79%.
The reason for a slight difference is that there still exists some disparity in the dataset provided to the model for training and testing. Gathering and training the model on a still

larger dataset would help overcome this issue and we would be able to achieve an even higher validation accuracy.

5.2 Rendering Engine/Application Development Framework

In terms of the application development framework we have used Flutter framework based on the Dart language developed by Google. Flutter is an open source SDK which can be used to develop cross platform application for IOS, Android,Windows, Mac, etc from a single code base. Single code essentially means that we don't require a separate code to develop apps for different platforms, The advantages of single code base are:

1. Implementaion becomes easy.
2. Same look and feel across all the Operating Systems.
3. Maintenance is easier.
4. Cost Efficient.

Fig 5.2.1 Flutter Framework

Above is the flutter framework where the flutter API uses the Dart coding language which can be compiled by software development kit to run on both android and iOS.

There are multiple frameworks available which uses single code base like React Native, Xamarin, Flutter, Iconic etc.Now the question arises is that out of all the frameworks at our disposal, why have we chosen Flutter? There are many reasons in selecting flutter :

1. Support by Google
2. Open Source
3. Dart as a Programming language
4. Community Support

We can also see the increase in trend of using Flutter as a framework for development in Fig 5.2.2

Now talking about the user interface that we have designed in our app.We have made the UI very user friendly, two buttons have been provided in the home screen for selecting the image which can be done either form the gallery or directly from the camera. After selecting one of these choices, the application first takes the permission from the user whether they want the app to allow in accessing the users gallery or camera.Granting permission is a mandatory part and the image the application won't move on to the image classification phase if the permission is denied from the user. The user can input any image using either methods and the result is displayed almost instantaneously on the screen indicating the calorie content of the food item identified by the model.

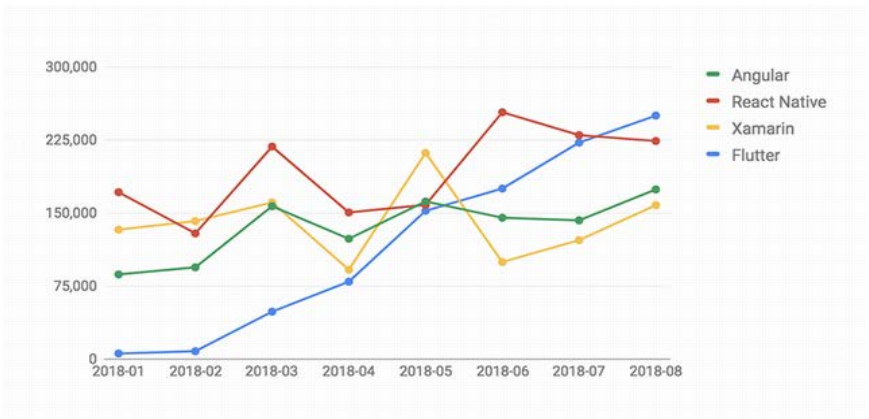

Fig 5.2.2 Increasing trend of Flutter

Fig 5.2.3 Frontend Design/ UI

In the image recognition phase, we have a function classify Image which receives image as an argument and gives the name of the selected food item as a result. This model is previously loaded by another function load Image Model which loads our trained model, The model here is converted into Tensorflow Lite (.tflite) , as

tensorflow lite model is suitable for small devices like mobile phones.

A profile page is also integrated in the application where user can maintain their profile which would contain data including their height, weight, calorie intake.
The mobile phone's local storage is used to store and fetch data for this purpose.

6. Results and Discussion

We have evaluated the performance of the models that we have trained on our dataset. Ou Dataset consists of 72 classes of different food items with around 300-500 images per class. In a nutshell we have about 22000+ images that the models are trained upon. Given below in Fig 6.1 are the training accuracy graphs of two major base architectures we have trained our dataset upon. For the part of training accuracy, InceptionV3 was lagging behind the ResNet50 architecture by a huge margin. InceptionV3 achieved a training accuracy of 62% and for ResNet50 it was 88% .We trained the model for 150 epochs. In the initial epochs both the architectures performed relatively well, but after 70-80 epochs the InceptionV3 achieved saturation and didn't had much of a growth, the reason could have been that the InceptionV3 largely focuses on the computational cost and also although in the initial epochs, accuracy increases, it saturates down in the later epochs due to overfitting. But for the case of ResNet50 they use skip connections that just takes out the case of Vanishing gradient out of the equation by skipping through irrelevant layers and thus giving opportunity to increase depth of the model by stacking more layers. Also in Fig 6.2 ResNet has better validation accuracy of 79% as compared to InceptionV3 which has 70% validation accuracy. The accuracy can be increased if we add more raw images to our dataset. Also improvement in augmentation strategy could yield a better result.

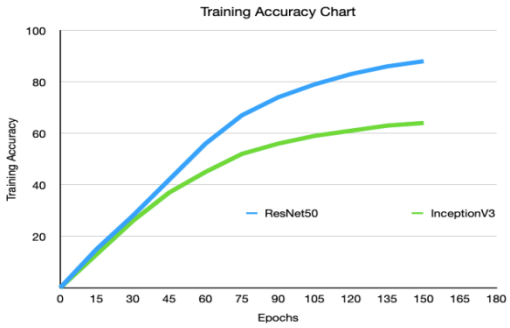

Fig 6.1 Training Accuracy Chart

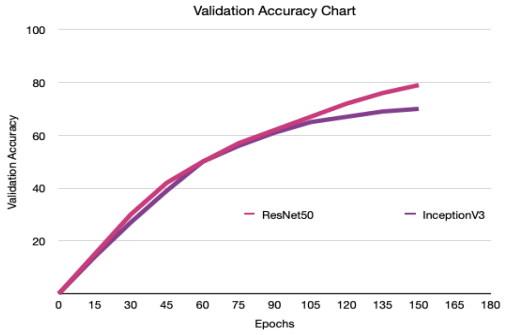

Fig 6.2 Validation Accuracy Chart

Also in Fig 6.3, training time of different models that we trained have been represented. The custom model was the initial model that we made from scratch by just stacking up some convolution layers, dense layers, etc and thus due to a shallow network it was trained in relatively small time ie 2hrs. ResNet50 on other hand was trained in about 18hrs for 150 epochs. It had 50 layers as its base and then we also stacked up some dropout layers and dense layers. The InceptionV3 has a training time of 20hrs and had 48 layers with some additional layers added manually.

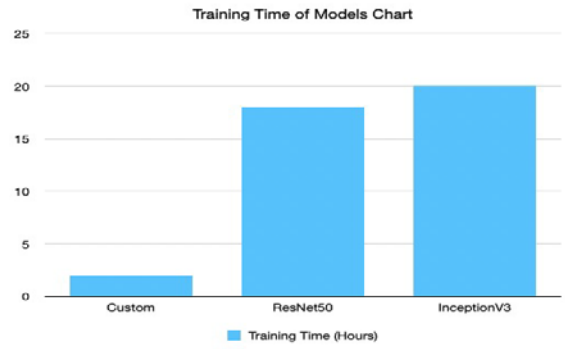

Fig 6.3 Training Time of different models

Below are the snapshots of the application. In Fig 6.4 the image of is captured after clicking the camera buttom. In Fig 6.5 the the image is classified and its name is displayed on the screen along with the calories associated to it.

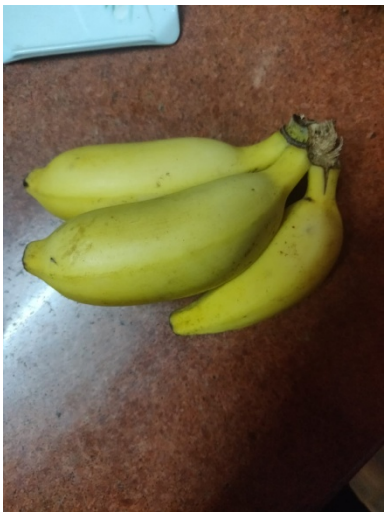

Fig 6.4 Image of Banana being captured

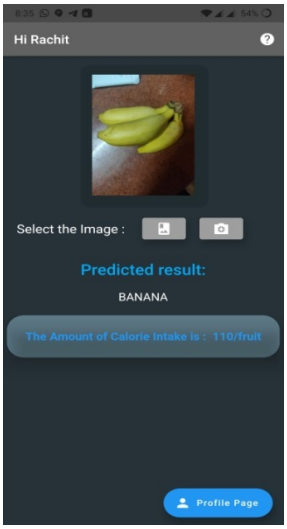

Fig 6.5 Model classifying the image and displaying the calories

7. Conclusion

Machine learning techniques previously used in the domain of Computer vision for Image recognition and detection task had a drawback that the feature extraction part of the process and also fine tuning of the paramenters was manually done which was a very cumbersome task. To eliminate these drawbacks, deep learning techniques are used and have performed drastically well in comparison to machine learning techniques. In this work, we have successfully trained renowned models like ResNet, InceptionNet and based on the performance statistics went ahead with the ResNet50 model. The frontend development of the android application was a novel task and was successfully completed using Flutter. The model performed accurately when tested over different food items.

8. Future Work

In future, we plan to work on improving the accuracy of our image recognition system using ResNets and also to reduce the processing time. We also aim at developing a multi-task food recognition system that identifies multiple food item in the image. Future work also includes further investigation on an optimal architecture and also using combination of machine learning classifiers with CNN features. Integrating our system with cloud computing and to scale our system from a calorie counter to a optimal diet predictor would be the things that we will strive upon.

9. References

[1] Chang Liu, Yu Cao, Yan Luo, Guanling Chen, Vinod Vokkarane and Yunsheng Ma (2014). DeepFood: Deep Learning-Based Food Image Recognition for Computer-Aided Dietary Assessment. In International Conference on Smart Homes and Health Telematics, pp 37-48

[2] Siddarth S, Sainath G and Vignesh S (2020). Deep Residual Network based food recognition for enhanced Augmented Reality application.

[3] Alex Krizehevsky, Ilya Sutskever and Geoffrey E Hinton(2012). ImageNet Classification with Deep Convolutional Neural Networks. In Conference on Neural Information Processing Systems (NIPS).

[4] L. Jiang, B. Qiu, X. Liu, C. Huang and K. Lin(2020). DeepFood: Food Image Analysis and Dietary Assessment via Deep Model, In IEEE Access, vol. 8, pp. 47477-47489, 2020, doi: 10.1109 ACCESS.2020.2973625

[5] Kaiming He, Xiangyu Zhang ,Shaoqing Ren, Jian Sun(2015). Deep Residual learning for Image Recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[6] Andrew Zisserman, Karen Simonyan(2015). Very Deep Convolutional Networks for Large Scale Image Recognition. In International Conference on Learning Representation(ILSR).

[7] Seon-Joo-Park, Chang-Ho Lee, Nanoom Jeong, Akmaljon Palmanov, Hae Jeung Lee(2019). The development of food detection and recognition model. In Nutrition Research and Practice.

[8] Stergios Christodoulidis, Marios Anthimopoulos , Stavroula Mougiakakou(2015). Food Recognition for Dietary Assessment Using Deep Convolutional Neural Networks. In International Conference on Image Analysis and Processing pp 458-465

[9] Zhang, X., Lu, Y. & Zhang, S(2016). Multi-Task Learning for Food Identification and Analysis with Deep Convolutional Neural Networks. J. Comput. Sci. Technol. 31, 489–500

[10] A. Şengür, Y. Akbulut and Ü. Budak(2019), Food Image Classification with Deep Features. International Artificial Intelligence and Data Processing Symposium (IDAP).

[11] Kuang-Huei Lee, Xiaodong He, Lei Zhang, Linjun Yang(2018). CleanNet : Transfer Learning for Scalable Image Classifier Training with Label Noise. Computer Vision and Pattern Recognition(CVPR).

[12] Ashutosh Singla, Lin Yuan, Touradj Ebrahimi(2016). Food/Non-food Image Classification and Food Categorization using Pre-Trained GoogLeNet Model. In MADiMa '16: Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management.