

PROFESSIONAL TRAINING REPORT

AT

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University)

Submitted in partial fulfilment of the requirements for the award of

Bachelor of Engineering Degree in

Information Technology

By

HARIHARAN V

(Reg no 40120034)



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119.

APRIL 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
(www.sathyabama.com)

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of HARIHARAN.V (Reg. No: 40120034) who carried out the project entitled " FOOD VISION USING TENSORFLOW " under my supervision from January 2023 to April 2023 .

R. n. Gov
Internal Guide

Dr. Gomathi, M.Tech., Ph.D.,

for R Subhashini
Head of the Department

Dr. R Subhashini, M.E., Ph.D.,

Submitted for Viva voce Examination held on 28/4/23

[Signature]
Internal Examiner

[Signature]
External Examiner

2023/10/11 17:26

DECLARATION

I, HARIHARAN. V hereby declare that the project report entitled " FOOD VISION USING TENSORFLOW " done by me under the guidance of Dr. R.M Gomathi., M.Tech., Ph.D., is submitted in partial fulfilment of the requirements for the award of Bachelor of Technology degree in Information Technology .

DATE: 28/4/23

PLACE: Chennai

Hariharan. V
SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D, Dean**, School of Computing **Dr. R Subhashini, M.E., Ph.D., Head of the Department** of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mrs. R.M. Gomathi., M.E., Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

ABSTRACT

According to some estimates of World Health Organization, in 2016, more than 1.9 billion adults were overweight. About 13% of the world's adult population were obese. 39% of adults were overweight. The worldwide prevalence of obesity more than doubled between 1980 and 2021. Nowadays, mobile applications recording food intake of people become popular. If an improved food classification system is introduced, users take the photo of their meals and system classifies photos into the categories. Hence, we proposed a convolutional neural network(CNN) structure trained on a pre-trained model efficientNetB04. Three different transfer learning based convolutional neural networks were trained using different learning methods:

- feature extraction with top 2 layers,
- fine-tuning with mixed precision training over 10 more layers unfreezed,
- fine-tuning with mixed precision training over 10 more layers unfreezed with an extra augmentation_layer over the input dataset.
- fine-tuning with mixed precision training with all layers unfreezed.

Train, validation and test datasets were generated from Food101 dataset. All tests were implemented through NVIDIA Server interface on K80s through Google's Colab. According to the test results, although pre-trained models provided better results than proposed structures, their performances were comparable. Moreover, learning optimization methods accelerated and improved the performances of all the compared models.

Keywords : Transfer learning, DCNN, food image recognition

TABLE OF CONTENTS:

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	vii
	LIST OF TABLES	viii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	
	1.1 INTRODUCTION	1
	1.2 OUTLINE OF THE PROJECT	2
2	AIM AND SCOPE OF THE PRESENT INVESTIGATION	
	2.1 AIM	3
	2.2 SCOPE	3
3	EXPERIMENTAL OR MATERIALS, METHODS AND ALGORITHMS USED	
	3.1 MODULES / METHODS	4
	3.2 EXTERNAL RESOURCES	5

	3.3 ALGORITHMS	7
4	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS	
	4.1 MODEL WORKING ARCHITECTURE	12
	4.2 PERFORMANCE ANALYSIS	12
5	SUMMARY AND CONCLUSIONS	
	5.1 SUMMARY AND CONCLUSION	14
	APPENDIX	15
	SOURCE CODE	
	REFERENCES	29

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
3.1	KINDS OF TRANSFER LEARNING AND THEIR WORKING ARCHITECTURE	8
3.2	A TENSORFLOW WORKFLOW	8
3.3	DATA AUGMENTATION LAYER	10
3.4	CHECK IF GPU IS MIXED PRECISION READY	10
3.5	CODE CHANGES TO USE MIXED PRECISION IN OUR MODEL TRAINING	10
4.1	OUR MODEL'S WORKING ARCHITECTURE	11
4.2	VALIDATION ACCURACY ON FOOD-101 AT DIFFERENT ITERATIONS	12

4.3	GRAPH FOR THE FINAL MODEL GENERATED IN TENSORBOARD	13
-----	---	----

LIST OF TABLES

TABLE NO.	TABLE NAME	PAGE N O.
4.1	COMPARISON OF ACCURACY ON FOOD-101 AT DIFFERENT ITERATIONS	12

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
CNN	Convolutional Neural Network
API	Application Programming Interface
Adam	ADaptive with Momentum
GPU	Graphics Processing Unit
GiB	Giga Byte
NLP	Natural Language Processing
TPU	Tensor Processing Unit

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION:

Accurate estimation of dietary caloric intake is important for assessing the effectiveness of weight loss interventions. Current methods for dietary assessment rely on self-report and manually recorded instruments (e.g., 24-hour dietary recall and food frequency questionnaires). Though the 24-hour dietary recall is the gold standard for reporting, this method still experiences bias as the participant is required to estimate their dietary intake (short and long term). Assessment of dietary intake by the participant can result in underreporting and underestimating of food intake. In order to reduce participant bias and increase the accuracy of self-report, enhancements are needed to supplement the current dietary recalls. One of the potential solutions is a mobile cloud computing system, which is to employ mobile computing devices (e.g., smartphone) to capture the dietary information in natural living environments and to employ the computing capacity in the cloud to analyse the dietary information automatically for objective dietary assessment. Among the large selection of mobile cloud computing software for health, many have proposed to improve dietary estimates. While these apps have features to track food intake, exercise, and save data in the cloud, the user has to manually enter all their information. To overcome these barriers, some research and development efforts have been made over the last few years for visual-based dietary information analysis. While progresses have been made, how to derive the food information (e.g., food type) from food image effectively and efficiently remains a challenging and open research problem. In this paper, we propose Transfer learning-based food image recognition to address this challenge. The proposed approach is based on Convolutional Neural Network (CNN) model EfficientNetB0 with a few major optimisations and . The experimental results of applying the proposed approach to a real-world datasets have demonstrated the effectiveness of our solution. The rest of the paper is organised as follows. Introduction to visual-based food recognition, details on the proposed Transfer learning-based approach for food image recognition, implementation details and the evaluation results of our proposed algorithms and finally I conclude with some remarks.

1.2 OUTLINE OF THE PROJECT:

- Using TensorFlow Datasets to download and explore data
- Creating preprocessing function for our data
- Batching & preparing datasets for modelling
- Creating modelling callbacks
- Setting up mixed precision training
- Building a feature extraction model
- Fine-tuning the feature extraction model
- Viewing training results on TensorBoard

CHAPTER 2

AIM AND SCOPE OF THE PRESENT INVESTIGATION

2.1 AIM:

In this work, I've implement a real-time food classification system using transfer learning using an existing pertained model (efficientNetB4). Using it as a base model for feature-extraction model, fine-tuning it to achieve the top accuracy with softmax activation and a learning rate of $\frac{1}{e^3}$ which is 10 times slower than normal learning rate.

2.2 SCOPE:

In the process of selecting the most suitable classification method for this scheme, several supervised algorithms were tested and compared in python using keras, a powerful open-source set of machine learning libraries that also includes pre-trained models and datasets. Space_categorical_crossentropy optimised by Adam optimizers is used to obtain more accurate performance results, measuring in each experiment:

- feature extraction with top 2 layers,
- fine-tuning with mixed precision training over 10 more layers unfreezed,
- fine-tuning with mixed precision training over 10 more layers unfreezed with an extra agumentation_layer over the input dataset.

Having a ModelCheckpoint to store the weights of the model with best validation accuracy.

- fine-tuning with mixed precision training with all layers unfreezed.

Has a special call back function to initiate early stop by monitoring validation loss with a patience of continual 3 losses.

CHAPTER 3

EXPERIMENTAL OR MATERIALS, METHODS AND ALGORITHMS USED

3.1 MODULES / METHODS :

3.1.a Tensorflow:

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

3.1.b Tensorflow_datasets:

TensorFlow Datasets is a collection of datasets ready to use, with TensorFlow or other Python ML frameworks, such as Jax. All datasets are exposed as `tf.data.Datasets`, enabling easy-to-use and high-performance input pipelines.

3.1.c Matplotlib:

An open-source Python 2D plotting library. matplotlib helps you visualize different aspects of machine learning.

3.1.d Tensorboard:

The dashboard that displays the summaries saved during the execution of one or more TensorFlow programs.

3.1.e tf.keras:

A popular Python machine learning API. [Keras](#) runs on several deep learning frameworks, including TensorFlow, where it is made available as [tf.keras](#).

3.1.f ModelCheckpoint:

Data that captures the state of the variables of a model at a particular time. Checkpoints enable exporting model weights, as well as performing training across multiple sessions. Checkpoints also enable training to continue past errors (for example, job preemption). Note that the graph itself is not included in a checkpoint.

3.1.g ReduceLROnPlateau:

Reduce learning rate when a metric has stopped improving.

3.1.h Adam:

Adam, which stands for ADaptive with Momentum. Implements Adam algorithm which automates the optimization using gradient over momentum.

3.1.i Softmax:

A function that provides probabilities for each possible class in a multi-class classification model. The probabilities add up to exactly 1.0. For example, softmax might determine that the probability of a particular image being a dog at 0.9, a cat at 0.08, and a horse at 0.02. (Also called full softmax.)

3.1.j Sparse_categorical_crossentropy:

Computes the crossentropy loss between the labels and predictions. Use this crossentropy loss function when there are two or more label classes. We expect labels to be provided as integers.

3.1.k EarlyStopping:

A method for regularization that involves ending model training *before* training loss finishes decreasing. In early stopping, you end model training when the loss on a validation dataset starts to increase, that is, when generalization performance worsens.

3.1.l SavedModel:

The recommended format for saving and recovering TensorFlow models. SavedModel is a language-neutral, recoverable serialization format, which enables higher-level systems and tools to produce, consume, and transform TensorFlow models.

3.2 EXTERNAL RESOURCES:

3.2.a Colab:

Colaboratory, or 'Colab' for short, allows you to write and execute Python / Swift in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

3.2.b Colab GPU:

The types of GPUs that are available in Colab vary over time. This is necessary for Colab to be able to provide access to these resources for free. The GPUs available in Colab often include Nvidia K80s, T4s, P4s and P100s. There is no way to choose what type of GPU you can connect to in Colab at any given time.

3.2.c Dataset:

This dataset consists of 101 food categories, with 101'000 images. For each class, 250 manually reviewed test images are provided as well as 750 training images. On purpose, the training images were not cleaned, and thus still contain some amount of noise. This comes mostly in the form of intense colors and sometimes wrong labels. All images were rescaled to have a maximum side length of 512 pixels.

- Homepage: https://data.vision.ee.ethz.ch/cvl/datasets_extra/food-101/
- Source code: `tfds.image_classification.Food101`
- Version: 2.1.0
- Download size: 4.65 GiB
- Extracted size: 6.5 GiB

3.2.d EfficientNetB0:

This TF-Hub module uses the Keras based implementation of EfficientNet-B0 . The default signature is used to classify images. Besides, the module contains a trained instance of the network, packaged to do the image classification that the network was trained on. We also offer a set of feature vectors to fit different downstream tasks.

- Homepage: <https://tfhub.dev/tensorflow/efficientnet/b0/classification/1>
- Source code: `tf.keras.applications.efficientnet.EfficientNetB0`
- Version: V5
- Download size: 19.26MB
- Format:TF2.0 Saved Model
- Fine Tunable: Yes
- License: Apache-2.0

3.2.e Graph from Tensorboard:

TensorBoard's Graphs dashboard is a powerful tool for examining your TensorFlow model. You can quickly view a conceptual graph of your model's

structure and ensure it matches your intended design. You can also view a op-level graph to understand how TensorFlow understands your program. Examining the op-level graph can give you insight as to how to change your model.

For example, you can redesign your model if training is progressing slower than expected.

3.3 ALGORITHMS:

3.3.a Transfer Learning:

Transferring information from one machine learning task to another. For example, in multi-task learning, a single model solves multiple tasks, such as a deep model that has different output nodes for different tasks. Transfer learning might involve transferring knowledge from the solution of a simpler task to a more complex one, or involve transferring knowledge from a task where there is more data to one where there is less data.

In other words, taking the patterns (also called weights) another model has learned from another problem and using them for our own problem.

There are two main benefits to using transfer learning:

- Can leverage an existing neural network architecture proven to work on problems similar to our own.
- Can leverage a working neural network architecture which has already learned patterns on similar data to our own. This often results in achieving great results with less custom data.

Different types of transfer learning are, as is, feature extraction and fine-tuning.

1. 'As is" transfer learning is when you take a pretrained model as it is and apply it to your task without any changes.
2. Feature extraction transfer learning is when you take the underlying patterns (also called weights) a pretrained model has learned and adjust its outputs to be more suited to your problem.

3. Fine-tuning transfer learning is when you take the underlying patterns (also called weights) of a pretrained model and adjust (fine-tune) them to your own

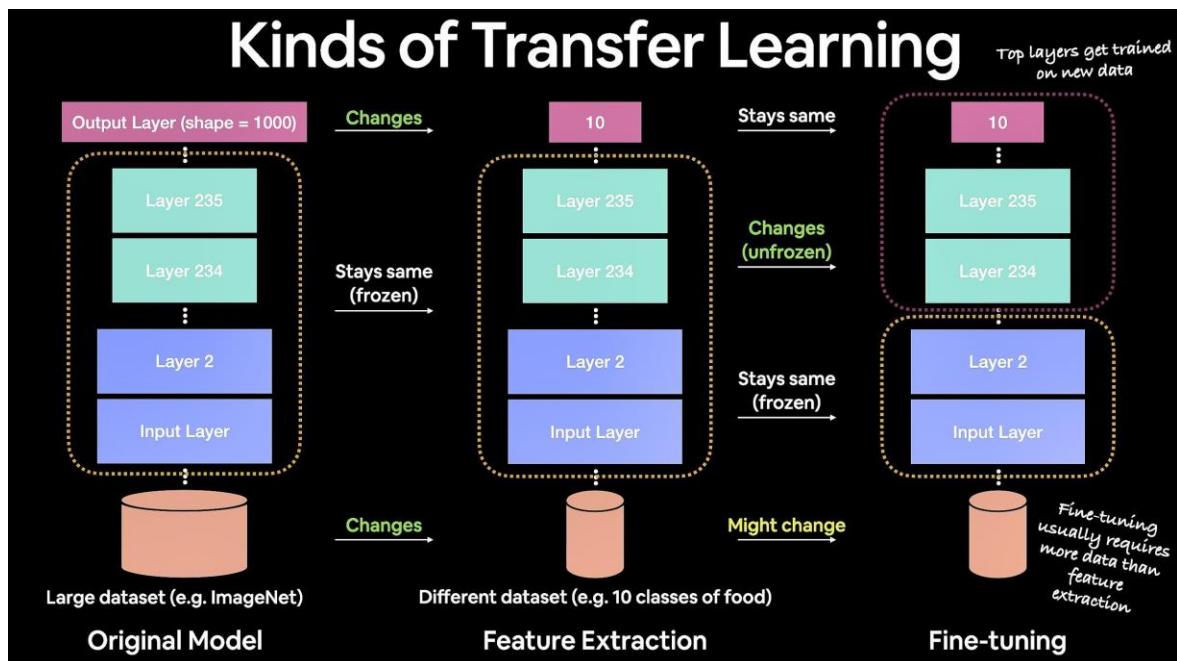


Fig 3.1 Kinds of Transfer Learning and their Working Architecture

problem.

What this means is, instead of hand-crafting our own neural network architectures or building them from scratch, we can utilise models which have worked for others. And instead of training our own models from scratch on our own datasets, we can take the patterns a model has learned from datasets such as ImageNet (millions of images of different objects) and use them as the foundation of our own. Doing this often leads to getting great results with less data.

Helpful if you don't have much data or vast compute resources. Use resources such as TensorFlow Hub or PyTorch Hub for broad model options and HuggingFace

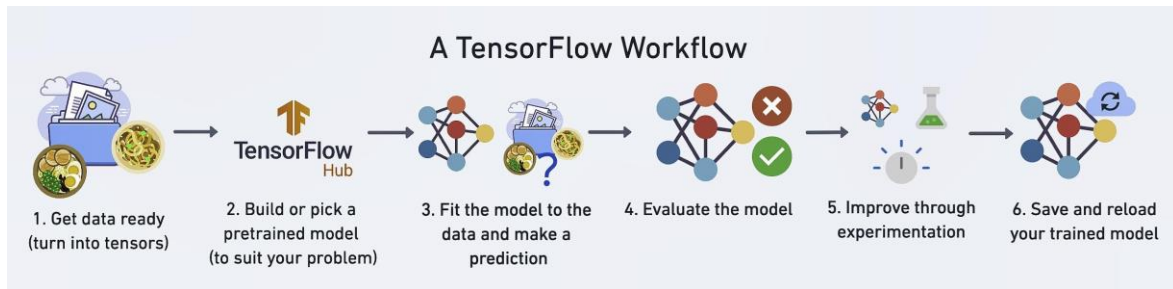


Fig 3.2 A TensorFlow workflow

transformers (NLP models) and Detectron2 (computer vision models) for specific models. Transfer learning is usually done for tasks where your dataset has too little data to train a full-scale model from scratch.

The most common incarnation of transfer learning in the context of deep learning is the following workflow:

1. Take layers from a previously trained model.
2. Freeze them, so as to avoid destroying any of the information they contain during future training rounds.
3. Add some new, trainable layers on top of the frozen layers. They will learn to turn the old features into predictions on a new dataset.
4. Train the new layers on your dataset.

3.3.b Data Augmentation:

Using the `tf.keras.layers.experimental.preprocessing` module a dedicated data augmentation layer can be created. Adding a data augmentation layer to the model has the following benefits:

- Preprocessing of the images (augmenting them) happens on the GPU rather than on the CPU (much faster).
- Images are best preprocessed on the GPU whereas text and structured data are more suited to be preprocessed on the CPU.
- Image data augmentation only happens during training so we can still export our whole model and use it elsewhere. And if someone else wanted to train the same model as us, including the same kind of data augmentation, they could.


```

mixed precision training
tf.keras import mixed_precision
mixed_precision.set_global_policy(policy="mixed_float16") # set global policy to mixed precision

tf.keras: Mixed precision compatibility check (mixed_float16): OK
You will likely run quickly with dtype policy mixed_float16 as it has compute capability of at least 7.0. Your GPU has compute capability 7.5

tf.keras: Mixed precision compatibility check (mixed_float16): OK
You will likely run quickly with dtype policy mixed_float16 as it has compute capability of at least 7.0. Your GPU has compute capability 7.5

```

Fig 3.4 Check if GPU is mixed precision ready

CHAPTER 4

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

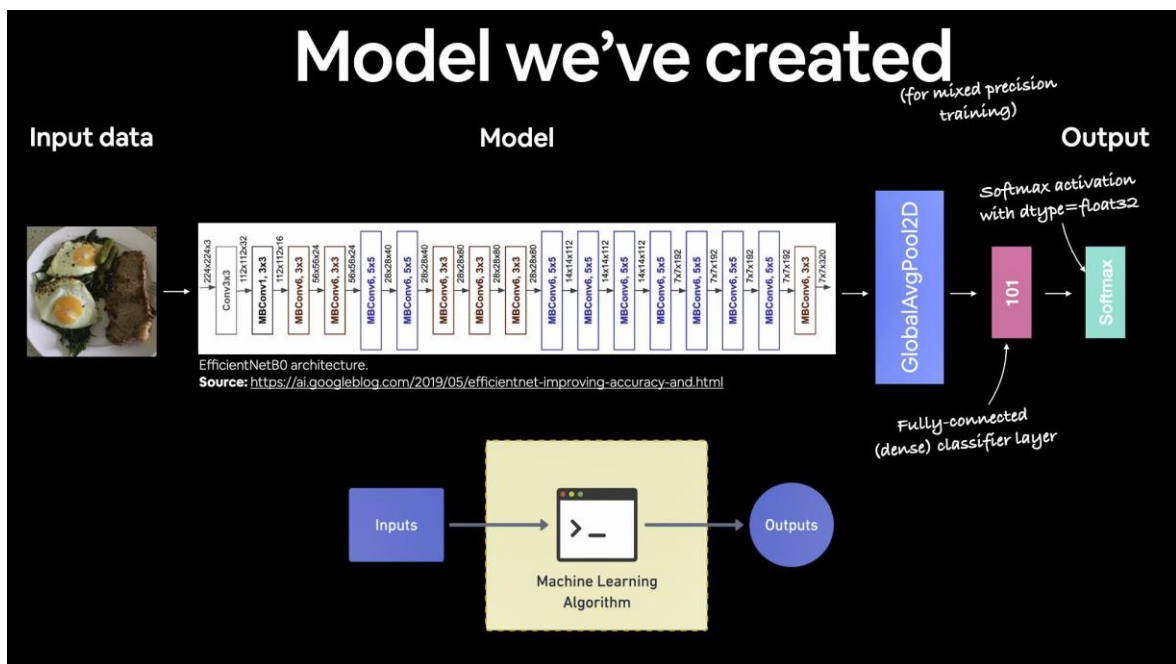


Fig 4.1 Our Model's Working Architecture

4.1 MODEL WORKING ARCHITECTURE:

4.2 PERFORMANCE ANALYSIS

The data set we used is Food-101 from tensorflow datasets. This dataset consists of 101 categories and each category has around 1000 images. Among the 1,000 images, around 75% of them were used for training and the rest 25% were used for testing. There are 101,000 images in total in this dataset. However, since all these data were collected by food sharing websites, images do not contain any bounding box information indicating the food location. Each image only contains the label information indicating the food type. Most of the images are popular western food images. The implementation of our algorithm for this dataset is similar to the one used in Fig 4.1. We did adjust the parameters to fit for 101 food categories, and then used a base learning rate of 0.0001, a momentum of 0.9 and 14210 iterations. Based on the 1000-class pre-trained model on ImageNet dataset, we fine-tuned the model on Food-101 dataset, the accuracy is shown as the following table, and we have achieved a 79.13% top-1 accuracy.

No of Iterations	Top-1 accuracy
2,000	75.05%
4,000	77.03%
6,000	78.08%
8,000	79.13%
10,000	78.04%
12,000	78.84%
14,000	78.8%
15,000	77.3%

Table 4.1 Comparison of accuracy on Food-101 at different iterations

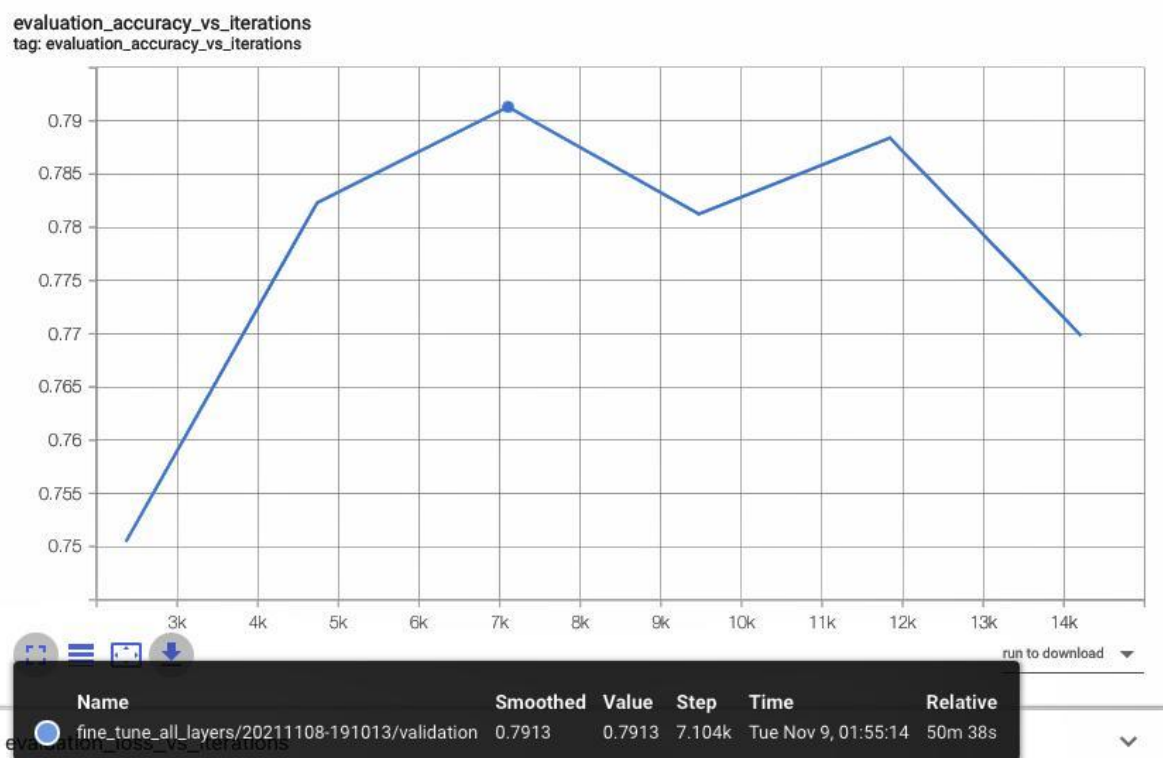


Fig 4.2 Validation accuracy on Food-101 at different iterations

SUMMARY AND CONCLUSION

5.1 SUMMARY AND CONCLUSION:

Obesity is a disorder involving excessive body fat that increases the risk of type 2 diabetes and cardiovascular diseases. In 2016, about 13% of the world's adult population (11% of men and 15% of women) were obese. Accurate estimation of dietary intake is important for assessing the effectiveness of weight loss interventions. In order to reduce bias and improve the accuracy of self-report, we proposed new algorithms to analyze the food images captured by mobile devices (e.g., smartphone). The key technique innovation in this paper is the Transfer Learning-based food image recognition algorithms using efficientNetB0 model . Our proposed model is based on Convolutional Neural Network (CNN). Our experimental results on two challenging data sets using our proposed approach exceed the results from all existing approaches.

In the future, we plan to improve performance of the algorithms and integrate our system into a realworld mobile devices and cloud computing-based system to enhance the accuracy of current measurements of dietary caloric intake.

colab.research.google.com

RAM

Disk

+ Code + Text Cannot save changes

[9]

)

Explore & visualized dataset

review some example

tfds.visualization.show_examples(train_data,ds_info)

pad_thai (70)

filet_mignon (37)

beef_carpaccio (3)

samosa (85)

huevos_rancheros (56)

lobster_bisque (60)

chicken_curry (18)

mussels (65)

omelette (67)

pad_thai (70)

filet_mignon (37)

beef_carpaccio (3)

samosa (85)

huevos_rancheros (56)

lobster_bisque (60)

chicken_curry (18)

mussels (65)

omelette (67)

[10] # Printout dataset detailed info

print('Number of class', ds_info.features["label"].num_classes)

print('Shape', ds_info.features.shape)

print('Dtype', ds_info.features.dtype)

Number of class 101

Shape {'image': (None, None, 3), 'label': {}}

Dtype {'image': tf.uint8, 'label': tf.int64}

Set class_names

[11] # Get class names

class_names = ds_info.features["label"].names

class_names

['apple_pie',

'baby_back_ribs',

'baklava',

'beef_carpaccio',

'beef_tartare',

'beet_salad',

5s completed at 3:52 AM

16

REFERENCES

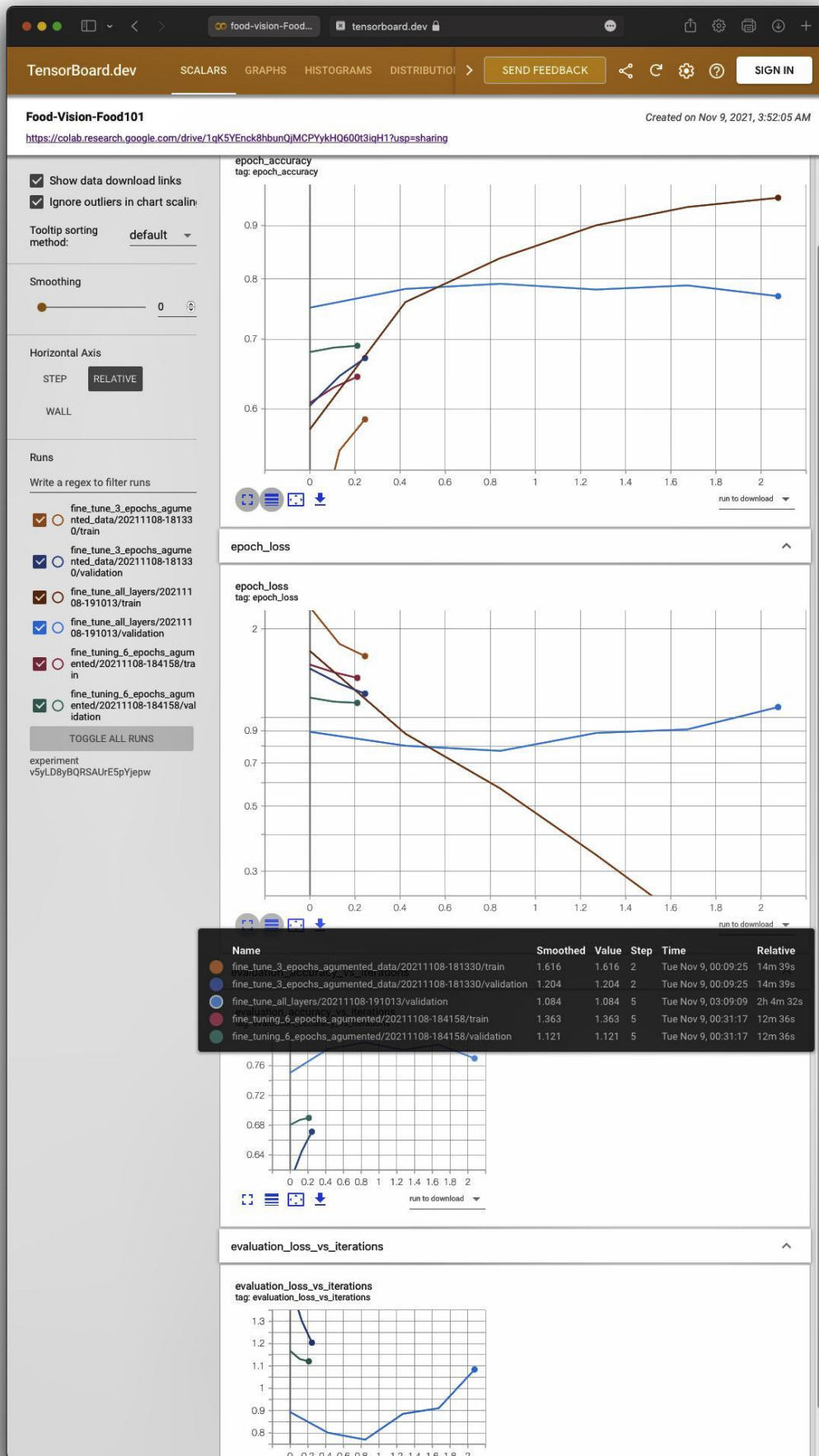
[1]Zhu, F., Bosch, M., Woo, I., et al: ‘The use of mobile devices in aiding dietary assessment and evaluation’, IEEE J. Sel. Top. Signal Process., 2010, 4, (4), pp. 756– 766 Crossref PubMed Web of Science®

Google Scholar

[2]Anthimopoulos, M., Gianola, L., Scarnato, L., et al: ‘A food recognition system for diabetic patients based on an optimized bag-of-features model’, IEEE J. Biomed. Health Inf., 2014, 18, (4), pp. 1261– 1271 Crossref PubMed Web of Science® Google Scholar

[3]Aizawa, K., Maruyama, Y., Li, H., et al: ‘Food balance estimation by using personal dietary tendencies in a multimedia food log’, IEEE J. Multimed., 2013, 15, (8), pp. 2176– 2185 Crossref Web of Science® Google Scholar

[4]Kitamura, K., de Silva, C., Yamasaki, T., et al: ‘Image processing based approach to food balance analysis for personal food logging’. 2010 IEEE Int. Conf. Multimedia and Expo (ICME), 2010, pp. 625– 630 Google Scholar



[5]Yanai, K., Takamu, T., Kawano, Y.: ‘Real-time photo mining from the

twitter stream: event photo discovery and food photo detection’. 2014 IEEE Int. Symp. Multimedia (ISM), 2014, pp. 295– 302 Google Scholar

[6]Matsuda, Y., Hoashi, H., Yanai, K.: ‘Recognition of multiple-food images by detecting candidate regions’. 2012 IEEE Int. Conf. Multimedia and Expo (ICME), 2012, pp. 25– 30 Google Scholar

[7]Dalal, N., Triggs, B.: ‘Histograms of oriented gradients for human detection’. IEEE Computer Society Conf. Computer Vision and Pattern Recognition 2005 CVPR 2005, 2005, vol. 1, pp. 886– 893 Google Scholar

[8]Lowe, D.G.: ‘Distinctive image features from scale-invariant keypoints’, Int. J. Comput. Vis., 2004, 60, (2), pp. 91– 110 Crossref Web of Science® Google Scholar

[9]Bossard, L., Guillaumin, M., Van Gool, L.: ‘Food-101 – mining discriminative components with random forests’. Computer Vision – ECCV 2014, 2014, pp. 446– 461 Google Scholar

[10]Zheng, J., Wang, Z.J., Zhu, C.: ‘Food image recognition via superpixel based low-level and mid-level distance coding for smart home applications’, Sustainability, 2017, 9, (5), p. 856 Crossref Google Scholar

[11]Li, Y., Liu, L., Shen, C., et al: ‘Mid-level deep pattern mining’. 2015 IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2015, pp. 971– 980Google Scholar

[12]Perronnin, F., Sánchez, J., Mensink, T.: ‘Improving the Fisher kernel for large-scale image classification’. Computer Vision – ECCV 2010, 2010, pp. 143– 156Google Scholar

[13]Bengio, Y., Courville, A., Vincent, P.: ‘Representation learning: a review and new perspectives’, IEEE Trans. Pattern Anal. Mach. Intell., 2013, 35, (8), pp. 1798– 1828CrossrefPubMedWeb of Science®Google Scholar

[14]Krizhevsky, A., Sutskever, I., Hinton, G.E.: ‘ImageNet classification with deep convolutional neural networks’, in: F. Pereira, C. Burges, L. Bottou, et al. (Eds.): ‘Advances in neural information processing systems 25’ (Curran Associates, Inc., 2012), pp. 1097– 1105. Available at <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>Google Scholar

- [15]Karpathy, A., Toderici, G., Shetty, S., et al: ‘Large-scale video classification with convolutional neural networks’. 2014 IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1725– 1732Google Scholar
- [16]Kawano, Y., Yanai, K.: ‘Food image recognition with deep convolutional features’. Proc. 2014 ACM Int. Joint Conf. Pervasive and Ubiquitous Computing: Adjunct Publication, 2014, pp. 589– 593 Google Scholar
- [17]Yanai, K., Kawano, Y.: ‘Food image recognition using deep convolutional network with pretraining and fine-tuning’. 2015 IEEE Int. Conf. Multimedia & Expo Workshops (ICMEW), 2015, pp. 1– 6 Google Scholar
- [18]He, Y., Xu, C., Khanna, N., et al: ‘Context based food image analysis’. 2013 20th IEEE Int. Conf. Image Processing (ICIP), 2013, pp. 2748– 2752Google Scholar
- [19]Farinella, G.M., Allegra, D., Moltisanti, M., et al: ‘Retrieval and classification of food images’, Comput. Biol. Med., 2016, 77, pp. 23– 39CrossrefPubMedWeb of Science®Google Scholar
- [20]Farinella, G.M., Moltisanti, M., Battiato, S.: ‘Food recognition using consensus vocabularies’. New Trends in Image Analysis and Processing – ICIAP 2015 Workshops, 2015, pp. 384– 392Google Scholar
- [21]Chen, M., Dhingra, K., Wu, W., et al: ‘PFID: Pittsburgh fast-food image dataset’. 2009 16th IEEE Int. Conf. Image Processing (ICIP), 2009, pp. 289– 292Google Scholar
- [22]Kawano, Y., Yanai, K.: ‘FoodCam: a real-time food recognition system on a smartphone’, Multimedia Tools Appl., 2015, 74, (14), pp. 5263– 5287CrossrefWeb of Science®Google Scholar
- [23]Doersch, C., Gupta, A., Efros, A. A.: ‘Mid-level visual element discovery as discriminative mode seeking’. Advances in Neural Information Processing Systems, 2013, pp. 494– 502Google Scholar
- [24]Martinel, N., Piciarelli, C., Micheloni, C.: ‘A supervised extreme learning committee for food recognition’, Comput. Vis. Image Underst., 2016, 148, pp. 67– 86CrossrefWeb of Science®Google Scholar
- [25]Hassannejad, H., Matrella, G., Ciampolini, P., et al: ‘Food image recognition using very deep convolutional networks’. Proc. Second Int. Workshop on Multimedia Assisted Dietary Management, 2016, pp. 41– 49Google Scholar

[26]Singla, A., Yuan, L., Ebrahimi, T.: ‘Food/non-food image classification and food categorization using pre-trained GoogleNet model’. Proc. Second Int. Workshop on Multimedia Assisted Dietary Management, 2016, pp. 3– 11Google Scholar

[27]Gong, Y., Wang, L., Guo, R., et al: ‘Multi-scale orderless pooling of deep convolutional activation features’. Computer Vision – ECCV 2014, 2014, pp. 392– 407Google Scholar

[28]Liu, L., Shen, C., Wang, L., et al: ‘Encoding high dimensional local features by sparse coding based Fisher vectors’. Advances in Neural Information Processing Systems, 2014, pp. 1143– 1151Google Scholar

[29]Felzenszwalb, P., Huttenlocher, D.P.: ‘Efficient graph-based image segmentation’, Int. J. Comput. Vis., 2004, 59, (2), pp. 167– 181CrossrefWeb of Science@Google Scholar

[30]Hinton, G.E.: ‘Deep belief networks’, Scholarpedia, 2009, 4, (5), p. 5947CrossrefGoogle Scholar

[31]Chen, M., Weinberger, K. Q., Sha, F., et al: ‘Marginalized denoising auto-encoders for nonlinear representations’. Proc. 31st Int. Conf. Machine Learning (ICML-14), 2014, pp. 1476– 1484Google Scholar

[32]Coates, A., Karpathy, A., Ng, A.Y.: ‘ Emergence of object-selective features in unsupervised feature learning’, in: F. Pereira, C. Burges, L. Bottou, et al. (Eds.): ‘ Advances in neural information processing systems 25’ (Curran Associates, Inc., 2012), pp. 2681– 2689. Available at <http://papers.nips.cc/paper/4497-emergence-of-object-selective-features-in-unsupervised-feature-learning.pdf>Google Scholar

[33]Oquab, M., Bottou, L., Laptev, I., et al: ‘Learning and transferring mid-level image representations using convolutional neural networks’. 2014 IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1717– 1724Google Scholar

[34]Oquab, M., Bottou, L., Laptev, I., et al: ‘Is object localization for free? – weakly-supervised learning with convolutional neural networks’. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2015, pp. 685– 694Google Scholar

[35]Feng, J., Wei, Y., Tao, L., et al: ‘Salient object detection by composition’. 2011 IEEE Int. Conf. Computer Vision (ICCV), 2011, pp. 1028– 1035Google Scholar

[36]Wang, X., Wang, B., Bai, X., et al: ‘Max-margin multiple-instance dictionary learning’. Proc. 30th Int. Conf. Machine Learning, 2013, pp. 846– 854Google Scholar

[37]Kawano, Y., Yanai, K.: ‘Automatic expansion of a food image dataset leveraging existing categories with domain adaptation’. Proc. ECCV Workshop on Transferring and Adapting Source Knowledge in Computer Vision (TASK-CV), 2014Google Scholar

[38]Long, M., Cao, Y., Wang, J., et al: ‘Learning transferable features with deep adaptation networks’. Int. Conf. Machine Learning, 2015, pp. 97– 105Google Scholar

[39]Jia, Y., Shelhamer, E., Donahue, J., et al: ‘Caffe: convolutional architecture for fast feature embedding’, arXiv preprint arXiv: