

TRANSFER LEARNING ON DEEP NEURAL NETWORK: A CASE STUDY ON FOOD-101 FOOD CLASSIFIER

Prakhar Tripathi
Computer science and engineering department
Chandigarh University
Punjab, India

Abstract—In the era of the ‘Big-Data’ we hear a lot about machine learning for working on this big data. Machine learning helps us to predict and analyze data with better accuracy and least human intervention. Machine learning is autonomous but susceptible to errors. This is due to biased prediction when previously trained on small data. This leads to chain of errors that can't be determined easily for long period of time. And when recognized takes lot time to recognize source.

There comes the idea of deep learning which achieves the flexibility by using use nested hierarchy of concept to define the world. But deep learning has setback of taking very long time to train data which could be reduced by using transfer learning.

Keyword- Deep neural network, Transfer learning, Food-101

I. INTRODUCTION

With the rise of nutrition related diseases a global awareness in consuming a balanced and healthy diet is growing. The consumption of healthy diets can reduce diseases like obesity, diabetes and cancer. Although there are wide number of applications already present to detect and classify food but all of them need prior knowledge to classify them. But the problem arises when a new and unknown food is introduced.

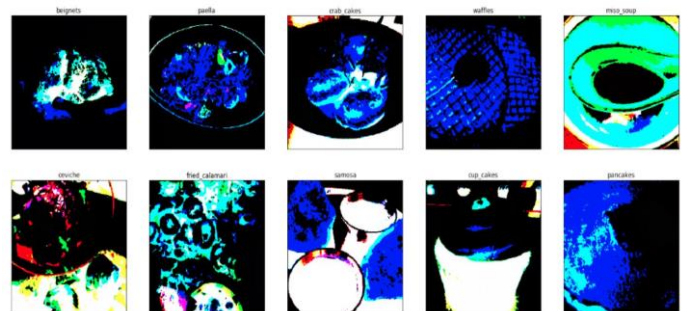
But this problem can now be solved using Convolutional Neural Network (CNN) and deep learning. However, there is still a lack of accurate results in the traditional methods of deep learning.

In response, we propose technique on pretrained **DenseNet-161** architecture and increasing class separability using successive augmentation of data before feeding it in model. In transfer learning, the knowledge of an already trained model is applied to various other problems. Transfer learning is used in area of computer vision to train machines and reduce learning cost. Most of the transfer learning architectures on CNN's architecture.

CNN have become powerful in image classification. Alexnet in 2012 ILSVRC had 65,000 neuron in 5 layers, followed by max-pooling layer and connected layer and then by softmax layer. DenseNet-161 has advantage over other

architecture because it uses less memory and computation which makes it suitable.

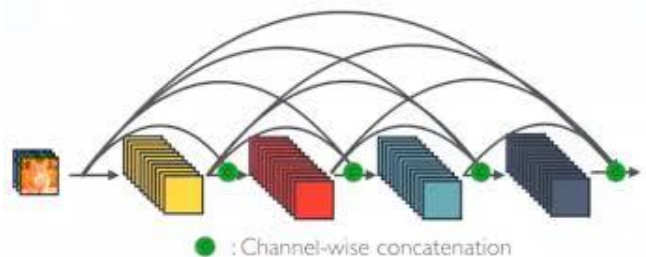
To evaluate the proposed architecture, experimental modelling has been done on benchmark data set (Food-101). And at the final results for Top-5 was found to be 99.01% accurate which is better than previous accuracy of 93.23% approximately.



Fig(1.1)-Food items in Food -101 data set

Food-101 data has various subset of datasets. The idea is to use down scaled version of images to enable quick test. The images have been reformatted using HDF5.

The proposed work is done using python programming language and pytorch library. After completion the results have been compared to other transfer learning architectures.



Fig(1.2)-DensesNet 161 architecture

II. RELATED WORK

A lot of scientific work is done on transfer learning architecture to increase accuracy in computer vision. Some

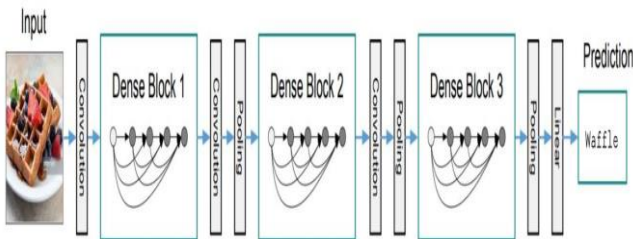
of the used approaches and their results are explained as: In Betawi, Traditional Food Image Detection using ResNet and DenseNet discussed on DenseNet 161(81%) and ResNet 50 with accuracy of approximately (76%).[6] Abdulkadir Şengür, Yaman Akbulut Ümit Budakin[2] in 2019 has also done work on pre-trained AlexNet and VGG16 on Food-5k, Food-101 and Food-11 data set. Lukas Bossard, Matthieu Guillaumin, Luc Van Gool[4] has worked on food-101 data also using random forests which gave accuracy of 50.76% .

III. MATERIAL AND METHODS

The main purpose of using a deep learning model along with transfer learning is to predict the images using computer vision with greater accuracy and lesser time and cost.

A. Proposed framework

This data has more than 100 thousand images which includes downgraded images [fig 1.1]. These have been reformatted using Keras HDF5. There has been use of convolutional layer followed by dense layer finally followed by pooling layer [fig 3.1]



IV. EXPERIMENT

A. DATASET

Food -101 dataset is used for this work which has 101,000 image data [4]. All images were re sized to 512 pixels. The images had a lot of noise for the purpose of work. The data is as food_c101_n1000_r384x384x3.h5 means there are 101 categories represented, with n=1000 images, that have a resolution of 384x384x3 (RGB, uint8) food_test_c101_n1000_r32x32x1.h5 means the data are part of the validation set, have 101 categories represented, with n=1000 images, that have a resolution of 32x32x1 (float32 from -1 to 1).

B. DATA PREPROCESSING

The dataset and the images in the raw state as obtained aren't suitable for further processing. Consequently, successive transformations are used to pre process the training dataset. In our implementation, these transformations include: Random rotation re-sized crop, Random horizontal flip, ImageNet policy and at the end Normalization.

These pre processing transforms are used to mitigate the disparities in image properties due to different image backgrounds; to help the model learn faster; and to improve the output accuracy

C. TRAIN AND TEST

To evaluate our model, we split the data set into training, test and validation in a ratio 8:1:1 i.e the 80% of the whole dataset was used for training and the rest equally split into test and validation.

D. MODEL USED

ResNet - In this architecture multiple layers are connected to one layer and all of them processed at once unlike other deep learning model.

Traditional feed-forward neural networks connect the output of the layer to the next layer after applying a **composite of operations**. This composite includes a convolution operation or pooling layers, a batch normalization and an activation function. The equation for this would be [3]

$$x_l = H_l(x_{l-1})$$

ResNets extended this behavior including the skip connection, reformulating this equation into: [5]

$$x_l = H_l(x_{l-1}) + x_{l-1}$$

DenseNets make the first difference with ResNets right here. DenseNets do not sum the output feature maps of the layer with the incoming feature maps but concatenate them.

E. ANALYSIS OF RESULTS-

Accuracy of results is done as

$$\% \text{ accuracy} = (\text{Correct data}/\text{Total data}) * 100$$

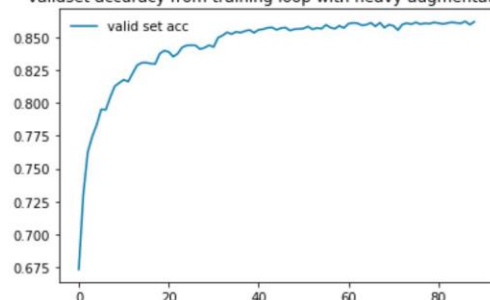
F. OPTIMIZATION OF RESULTS-

To improve our model classification accuracy and reduce the derived error values, we fine tuned the network parameters using Adam optimizer.

V. RESULT AND DISCUSSION

After training, validating and testing results found showed that DenseNet-161 showed great and accurate results. The results show better performance with respect to existing approaches. Specifically, we obtained a Top-1 accuracy of 93.27% and Top-5 accuracy around 99.02% on the Food-101 dataset). The results showed better results than any other model present and other algorithms.

validset accuracy from training loop with heavy augmentations





Method	Top - 1	Top - 5	Publication
HoG	8.85	-	ECCV2014
SURF BoW-1024	33.47	-	ECCV2014
SURF IFV-64	44.79	-	ECCV2014
SURF IFV-64 + Color Bow-64	49.40	-	ECCV2014
IFV	38.88	-	ECCV2014
RF	37.72	-	ECCV2014
RCF	28.46	-	ECCV2014
MLDS	42.63	-	ECCV2014
RFDC	50.76	-	ECCV2014
SELC	55.89	-	CVIU2016
AlexNet-CNN	56.40	-	ECCV2014
DCNN-FOOD	70.41	-	ICME2015
DeepFood	77.4	93.7	COST2016
Inception V3	88.28	96.88	ECCV2016
ResNet-200	88.38	97.85	CVPR2016
WRN	88.72	97.92	BMVC2016
ResNext-101	85.4	96.5	Proposed
WISeR	90.27	98.71	UNIUD2016
DenseNet - 161	93.26	99.01	Proposed

VI. CONCLUSION AND FUTURE WORK

The results given by DenseNet-161 show very accurate results with greater speed of performance in GPU. This model is using many number of nodes making it very good performer when we use 10 epochs but this method when used on CPU can be a bit slow. The current result presented results than its previous best accuracy of 93.27% on food 101 data. Also it out performed simple model like random forest method (50.76%).

Future scope for better results can be searched by developing other methods of transfer learning just like mobile net which is very light to work on but still lacks in accuracy issues. So in total we could create new models which has great performance and accuracy on mediocre devices. Further, performance can be improved using high end computers with GPU's and other mediums. Also there can be improvements by increasing the quality and quantity of data sets. This can also be done by increasing the number classes in Food -101 data, which could finally give results on other characteristics of objects. The results in future can also be improved by the creation and innovation in field of mathematics to find newer equations.

VII. REFERENCES

- [1] Huang Gao, Liu Zhuang, Weinberger Kilian Q., Maaten Laurens Van der ;2017; Dense Convolutional Network (DenseNet) [arxiv 1608.0699] The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4700-4708
- [2] Şengür Abdulkadir, Akbulut Yaman , Budakin Umit; 2019; Food Image Classification with Deep Features; Computer Science International Artificial Intelligence and Data Processing Symposium (IDAP)
- [3] He K., Zhang X., Ren S., and Sun J.; 2015; Deep Residual Learning for Image Recognition; arXiv:1512.03385v1
- [4] Bossard Lukas , Matthieu Guillaumin, Gool Luc Van ; 2014; Food-101 – Mining Discriminative Components with Random Forests; European Conference on Computer Vision Computer Vision – ECCV 2014 pp 446-461
- [5] Huang G., Liu Z. and Maaten L. van der; 2018; Densely Connected Convolutional Networks; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4700-4708
- [6] Setyono Noer Fitria Putra, Cahyati Dina , Fanany Mohamad Ivan; 2018; Betawi Traditional Food Image Detection using ResNet and DenseNet; IEEE 978-1-7281-0135-4/18/\$31.00 INSPEC Accession Number 18396749
- [7] Kumar R., Gredd K. and Schmidhuber J.; 2015; Highway Networks; arXiv:1507.06228
- [8] Baoqi Li and Yuyao He ; An Improved ResNet Based on the Adjustable Shortcut; 2018; IEEE Electronic ISSN: 2169-3536

```

accuracy += torch.mean(equals.type(torch.FloatTensor))
print(valid_loss/len(test_loader))
print(accuracy/len(test_loader))

Top-5 Acc.

valid_loss += valid_loss.item()
top_p, top_class = logs.topk(5, d)
#print(top_class)
#print(labels)
if (labels in top_class):
    equal=equal+1
#print(equal)
#equals = top_class == labels.view(
#accuracy += torch.mean(equals.type(
print(valid_loss/len(test_loader))
print(equal/len(test_loader))

Top-1 Acc.

0.23865746426163242
0.9326171875

In [ ]: model.load_state_dict(f
In [8]: valid_loss=0
accuracy=0
with torch.no_grad():
    model.eval()
    for i, (inputs, labels) in enumerate(test_loader):
        outputs = model(inputs)
        _, predicted = outputs.max(1)
        equals = predicted == labels
        accuracy += torch.mean(equals.type(torch.FloatTensor))
    print(accuracy/len(test_loader))
    print(valid_loss/len(test_loader))
    print(equal/len(test_loader))
    
```



- [9] Ari Nugroho, Heru Suhartanto;2020; Hyper-Parameter Tuning based on Random Search for DenseNet Optimization, Information Technology Computer and Electrical Engineering 2020 7th International Conference on, pp. 96-99.
- [10] Puteri Khatya Fahira, Ari Wibisono, Hanif Arief Wisesa, Zulia Putri Rahmadhani, Petrus Mursanto, Adi Nurhadiyatna;2019;Sumatra Traditional Food Image Classification Using Classical Machine Learning; Informatics and Computational Sciences (ICICoS) 2019 3rd International Conference on, pp. 1-5.
- [11] Enhui Lv, Xuesong Wang, Yuhu Cheng, Qiang Yu;2018;Deep Convolutional Network Based on Pyramid Architecture; Access IEEE, vol. 6, pp. 43125-43135.