[[1]](#footnote-1)

Multiclass Image Classification using convolutional Neural Network

*Abstract*—Convolutional Neural Networks (ConvNets) are commonly developed at a fixed resource budget, and then scaled up for better accuracy if more resources are available. In this paper, we systematically study model scaling and identify that carefully balancing network depth, width, and resolution can lead to better performance. Based on this observation, we propose a new scaling method that uniformly scales all dimensions of depth/width/resolution using a simple yet highly effective compound coefficient. We demonstrate the effectiveness of this method on scaling up

Nomenclature

# Introduction

Scaling up ConvNets is widely used to achieve better accuracy. For example, ResNet can be scaled up from ResNet-18 to ResNet-200 by using more layers; Recently, GPipe achieved 84.3% ImageNet top-1 accuracy by scaling up a baseline model fourFor such multiple case scenarios we can use Single Variable Linear Regression to predict the outcome of sale price among all the circumstance and draw out a full pledged inference form our learning (model).

Rajat Maheshwari, *Student, LPU (11807504)*

[*Github\_Link*](https://github.com/Rajat72/deep_learning)

However, the process of scaling up ConvNets has never been well understood and there are currently many ways to do it. The most common way is to scale up ConvNets by their depth (He et al., 2016) or width (Zagoruyko & Komodakis, 2016). Another less common, but increasingly popular, method is to scale up models by image resolution (Huang et al., 2018). In previous work, it is common to scale only one of the three dimensions – depth, width, and image size. Though it is possible to scale two or three dimensions arbitrarily, arbitrary scaling requires tedious manual tuning and still often yields sub-optimal accuracy and efficiency.

# Proposed Method

If the goal is speed, image processing, or classification CNN can be used to fit a classifying model to an observed [data set](https://en.wikipedia.org/wiki/Data_set) of values of the response and explanatory variables. After developing such a model, if additional values of the explanatory variables are collected without an accompanying response value, the fitted model can be used to make a classification of the response.

## Literature Review

|  |  |  |
| --- | --- | --- |
| Sno | Paper | year |
| 1 | EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks | 2020 |
| 2 | Food-101 – Mining Discriminative Components with Random Forests | 2014 |
| 3 | A survey of image classification methods and techniques for improving classification performance | 2007 |
| 4 | A Survey on Transfer Learning | 2010 |
| 5 | Resnet in Resnet: Generalizing Residual Architecture | 2016 |

## Exploratory Data Analysis(dataset)

A lot of data needs preprocessing before it is fed to any model.

Namely dealing with image processing and banding them in correct groups [link](https://www.tensorflow.org/datasets/catalog/food101)

Examples of classes:-



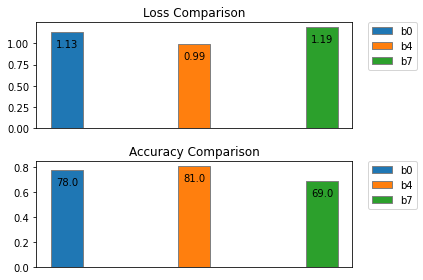
Random sample from dataset:-



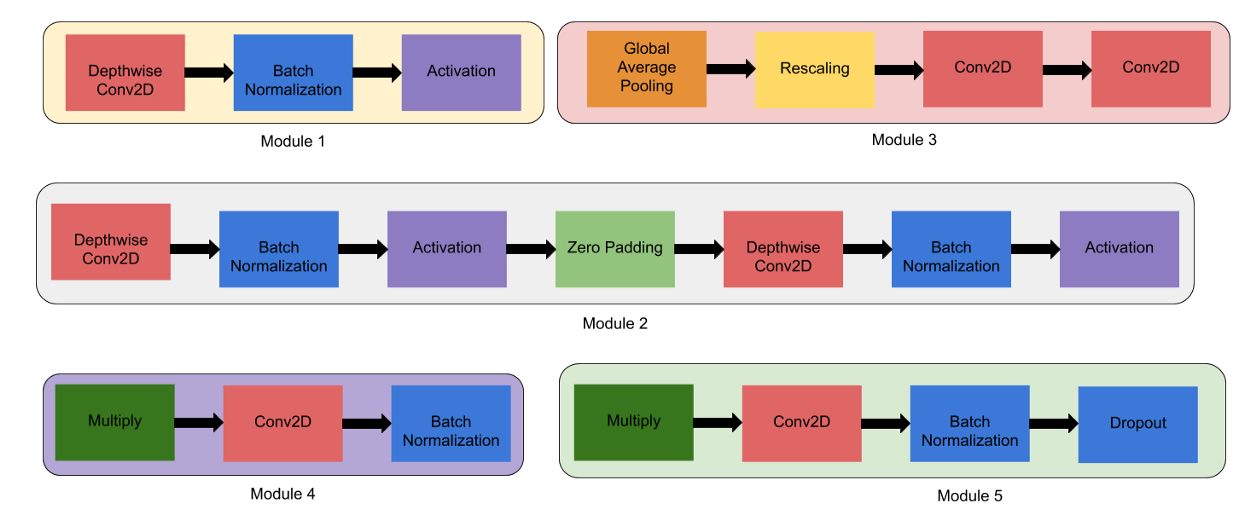
## Model comparison(result)

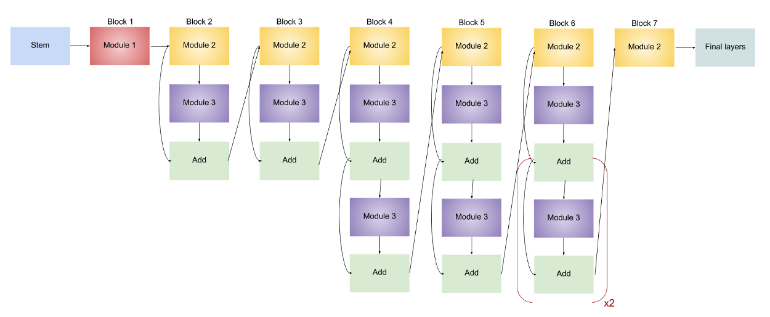
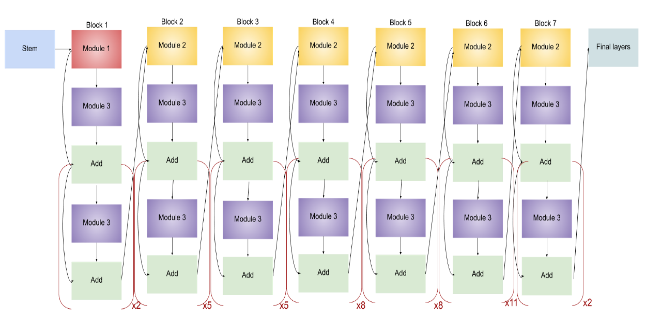
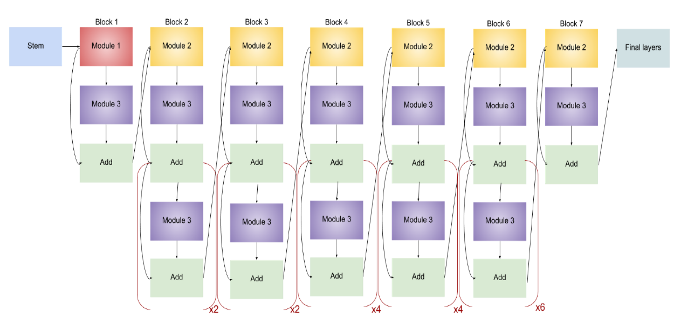
TABLE I

|  |  |  |
| --- | --- | --- |
| **Models** | **Loss** | **Accuracy (%)** |
| B0 | 1.13 | 78 |
| B4 | 0.99 | 81 |
| B7 | 1.19 | 69 |

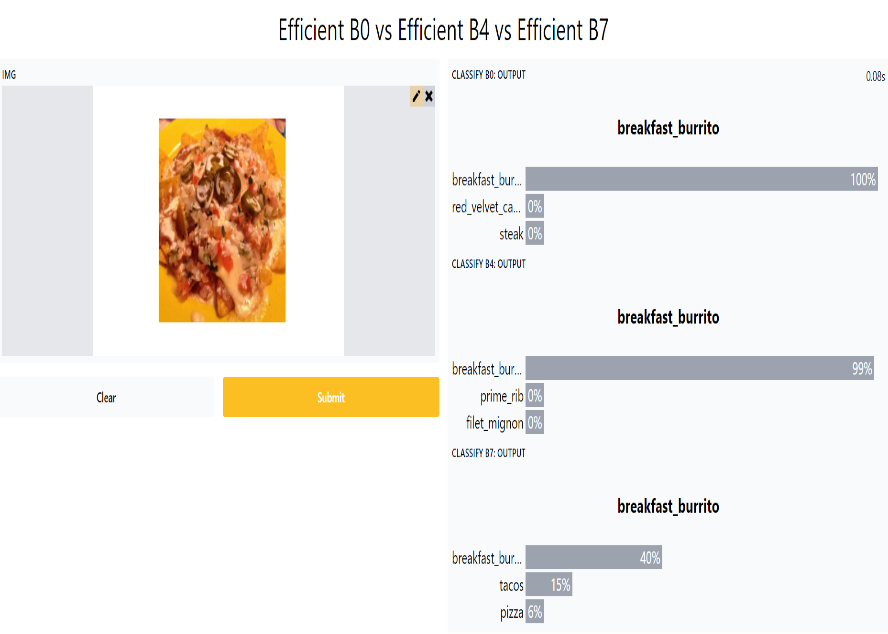


## Proposed Architecture



1. B0:- 
2. B4:- 
3. B7:- 

## Project(Screenshot)

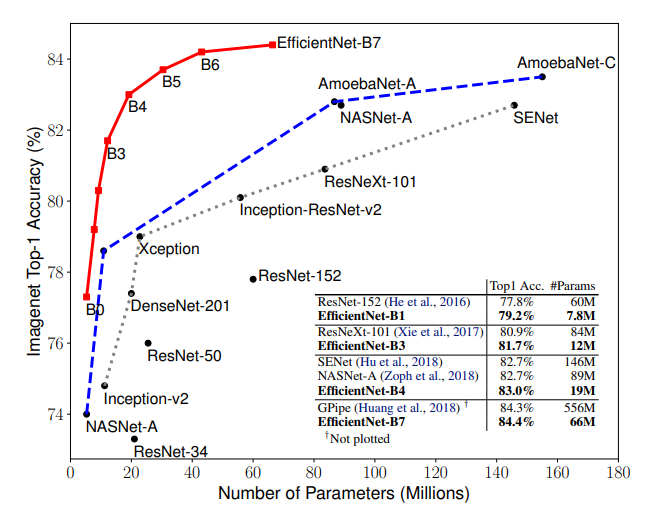


## Conclusion and future Scope

EfficientNetB0 is the fastest in terms of training.

EfficientNetB4 is the most efficient overall.

EfficientNetB7 has most parameters to train.

These transfer learning techniques can be implemented in other domain as well while scaling previous existing models like mentioned above 

# References

1. [Complete Architectural Details of all EfficientNet Models](https://towardsdatascience.com/complete-architectural-details-of-all-efficientnet-models-5fd5b736142)

1. [↑](#footnote-ref-1)