# Analysis of CNNs used in Image Recognition

## CS2IS2 Final Project (2018/2019)

Swastik Sahu, Sahir Sharma

sahus@tcd.ie, sharmas@tcd.ie

**Abstract.** Image classification has always been a challenging problem. Since 2012, significant advancements have been made in this area. In 2012, the winner of ImageNet challenge built a Deep Neural Network (DNN) which was better than all other models by a huge margin. That gave rise to more research in using DNNs for image classification. A series of state of art DNN architectures have been published since then. This paper presents a brief analysis of these DNN architectures.

## 1 Introduction

Image recognition is a challenging problem for the computer. Even recognising simple objects, for example - digits or letters can be an exceedingly complex task for the computer. An average human can correctly recognise almost all the digits in *figure 1* as the digit 'eight' with just a glance. This is a very challenging problem for a computer, and a simple logic may not even achieve an accuracy of 50%. Perhaps one of the few things humans can do more efficiently than computers!

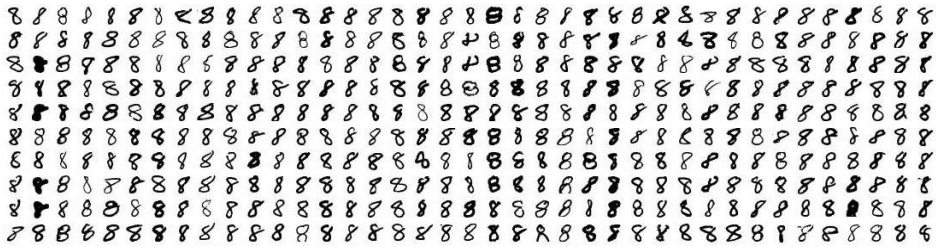

**Figure 1**

Unlike humans, computers do not perceive and classify the input as a certain type of shapes or figure. The input received by the computer is in the form of a series of pixel values, which may look like the example in *figure 2.* The slightest of change in the input, in the form of colour, hue, saturation, etc., can completely change the input. The complexity only adds up when this must be done for high-resolution images and/or in real-time situations like for autonomous vehicles.

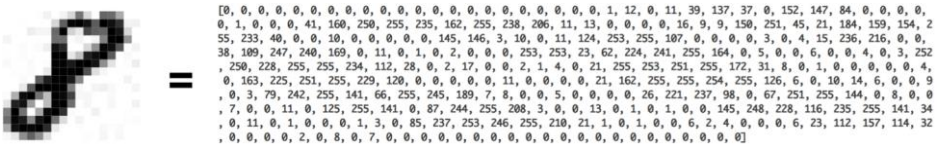


**Figure 2**

Till 2011, researchers have tried to solve this problem using various machine learning algorithms like Support Vector Machine, k-Nearest Neighbour and other classification algorithms, and were able to bring down to error rate to around 26%. The year 2012 marked the beginning of a new era, when researchers Krizhevsky et al. 2012 [1] from the University of Toronto, Canada used Deep Neural Networks to solve this challenge and were able to bring down the error rate to around 15%, a significant improvement over the previous best. Since then, the use of Deep Neural Networks in image recognition has been very popular and the current state of the art is as good as a human.

Deep Convolutional Neural Networks (DNN/CNN) came into existence around 1980 when researchers were trying to model the human brain. But its popularity faded away as it was not able to produce promising results. At that time, we did not have enough computing resources because of which researchers could not use it for practical purposes. In the last decade, the availability of computing resources has facilitated the used of DNN and it has regained its popularity. The architectural differences and performance analysis of state of art DNNs for image recognition, as they have evolved, have been discussed in this paper.

## 2 Literature Review

Krizhevsky et al. 2012 [1] from the University of Toronto, Canada developed a DNN (called AlexNet) which had 60 x $10^6$ parameters, 6.5 x $10^2$ neurons, with 5 convolutional layers followed by max-pooling layers, and 3 fully-connected layers with 1000-way softmax for image classification. Their model won the ImageNet competition by a huge margin and drew the attention of others researcher for using DNNs in image classification tasks.

In 2014, Zisserman and Simonyan [2] from the Visual Geometry Group (VGG), University of Oxford, modified the AlexNet to build an even deeper CNN called VGGNet. VGGNet used very small(3x3) convolution filter with a network of increasing depths, pushing the depth to 16-19 weight layers. Their model won the ImageNet 2014 challenge and achieved state of art results on other datasets.

Kaiming et al. (2015) [3] build a residual learning framework which makes training deeper CNNs less complex and were able to build an ensemble of residual nets with 152 layers. Their model (RESNet) achieved an error rate of 3.57% on the ImageNet 2015 test set and secured the first position for the classification task. The authors [4] further optimized the model by using identity mappings as the skip connections and after-addition activation to propagate information across the residual building blocks. This made training easier and improved generalization.

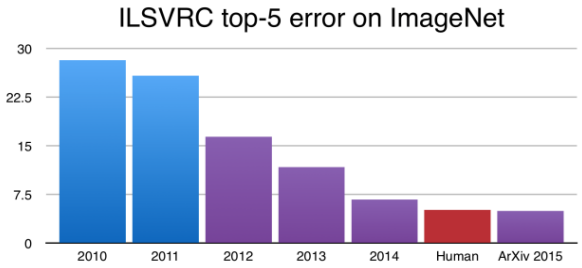


**Figure 3**

Xie et al. (2016) [5] further improved the ResNet model by aggregating residual transformations within the same topology. Their model, named ResNeXt, secured 2nd place in the 2016 ImageNet challenge.

Szegedy et al. (2015) [6] were the first to address the computational challenges by using Hebbian principal and mulit-scale processing. The model, aka GoogLeNet, uses an inception module and is the first to introduce the idea of increasing width of the network without losing on accuracy. Inception-ResNet [7] and Xception [8] are other models which were built as improvements over GoogLeNet.

Huang et al. (2016) [9] proposed a 'densely' connected CNN, called DenseNet, which connects a layer to every other layer in the model by propagating the feature map of all previous layers to the next layer. DenseNet obtained significant improvements over most of the state of the art while using less computational resources. It addresses the vanishing-gradient problem by strengthening feature propagation and encouraging feature reuse. The number of parameters required is also substantially reduced.

Zoph et al. (2017) [10] have proposed a transferable learning architecture for scalable image recognition, called NASNets. In their approach, they look for the best convolution layer (or cell) on a smaller search space (dataset) and apply more of these cell(s) to a larger dataset. Their approach is computationally demanding but surpasses all previously achieved accuracy levels.

## 3 Problem Definition

A standard way of evaluating the performance of different algorithms is necessary. There are several labelled datasets from CIFAR, ImageNet, PASCAL, and others that are available for this task.

The CIFAR (Canadian Institute for Advanced Research) dataset has around 60,000 tiny images (32x32 pixel), with 10/100 object classes.

PASCAL Visual Object Classification (PASCAL VOC) challenge ran from 2005-2012. PASCAL dataset had around 20,000 images with 20 object classes.

ImageNet is the most popular image classifying competition since 2010. It consists of over 1 Million high-res labelled training data which are categorized into 1000 groups. Instead of having simple classes - like 'dog', they are labelled with more detail, like – 'German Shephard'.

The ImageNet dataset is considered as the gold-standard for image classification problems. In this paper, selected models of the state-of-the-art techniques are used to classify a subset (3500 images) of images from the validation set of the ImageNet dataset (ILSVRC2012).


**Figure 4**

## 4 Deep Neural Network Architectures

Key state-of-the-art DNNs used in this analysis have discussed very briefly in this section.

### 4.1 AlexNet Architecture

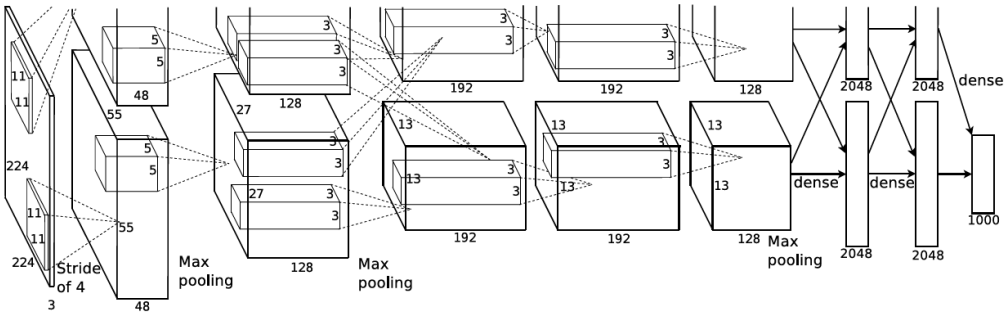


**Figure 5**

AlexNet was the first of the DNNs to outperform other classifiers. It is relatively shallow compared to the other DNN architectures.

## 4.2 VGGNet Architecture

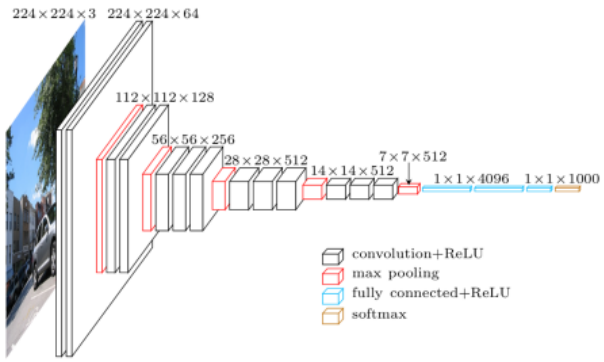


**Figure 6**

## 4.3 Inception Module from GoogLeNet

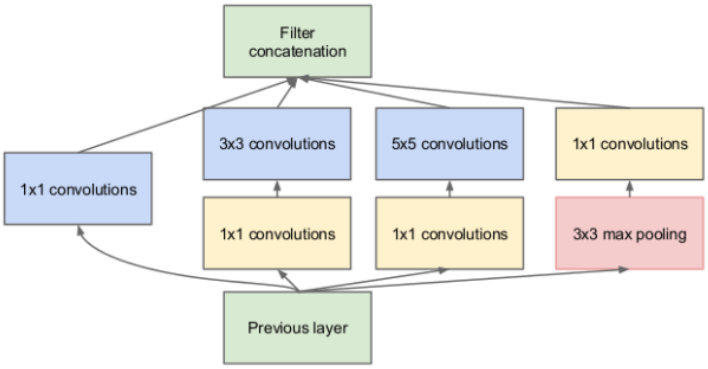


**Figure 7**

## 4.4 Identity mapping in ResNet Architecture

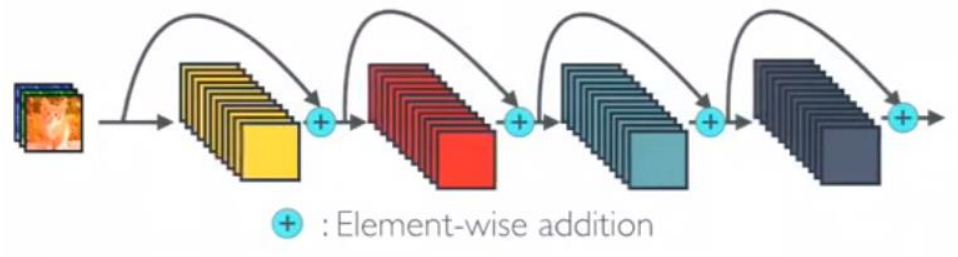


**Figure 8**

### 4.5 One Dense block DenseNet Architecture

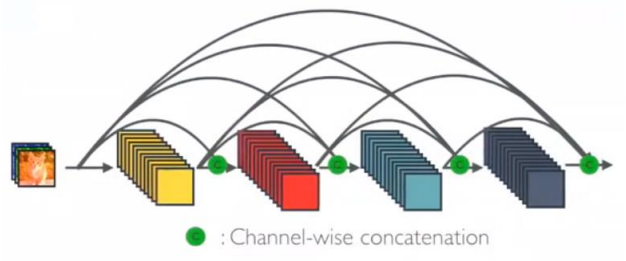


**Figure 9**

## 5 Experimental Results

### 5.1 Methodology

It takes GPU days to train DNN with a large amount of data. The inventors of these architectures have been generous enough to publicly share the weights of their trained models. The models are also available in various frameworks.
Keras is one such API in TensorFlow which provides high level implementations of these models. The Keras module has been used to process 3500 images taken the validation set of ImageNet 2012's dataset. The outputs of the neural networks are produced in the form of class probabilities. If the input is in the 'top-k' predictions, then it is considered correct for that 'top-k' group. Results are discussed in the next section.
The code is present in this repository: https://github.com/sahirsharma/CS7IS2_Assignment2_Repo

### 5.2 Results

*Table 1* presents the top-k accuracy scores, where k = [1, 10], for prediction models used in the experiment. Same data is presented in the form of a graph in *figure 10*.

| | Top-1 | Top-2 | Top-3 | Top-4 | Top-5 | Top-6 | Top-7 | Top-8 | Top-9 | Top-10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **VGG16** | 0.7 | 0.81 | 0.86 | 0.89 | 0.9 | 0.91 | 0.92 | 0.93 | 0.93 | 0.94 |
| **VGG19** | 0.7 | 0.82 | 0.86 | 0.89 | 0.9 | 0.91 | 0.92 | 0.92 | 0.93 | 0.93 |
| **RESNET50** | 0.75 | 0.85 | 0.89 | 0.91 | 0.92 | 0.93 | 0.94 | 0.95 | 0.95 | 0.95 |
| **DENSENET121** | 0.75 | 0.86 | 0.9 | 0.92 | 0.93 | 0.94 | 0.94 | 0.95 | 0.95 | 0.96 |
| **DENSENET169** | 0.76 | 0.87 | 0.91 | 0.93 | 0.94 | 0.95 | 0.95 | 0.96 | 0.96 | 0.96 |
| **DENSENET201** | 0.77 | 0.87 | 0.9 | 0.93 | 0.94 | 0.95 | 0.95 | 0.95 | 0.96 | 0.96 |
| **INCEPTIONV3** | 0.78 | 0.88 | 0.91 | 0.92 | 0.94 | 0.95 | 0.95 | 0.95 | 0.96 | 0.96 |
| **INCEPTION_RESNET_V2** | 0.79 | 0.9 | 0.93 | 0.94 | 0.95 | 0.96 | 0.97 | 0.97 | 0.97 | 0.98 |
| **XCEPTION** | 0.79 | 0.89 | 0.92 | 0.94 | 0.95 | 0.95 | 0.96 | 0.96 | 0.97 | 0.97 |
| **NASNET** | 0.82 | 0.91 | 0.94 | 0.96 | 0.96 | 0.97 | 0.97 | 0.97 | 0.98 | 0.98 |

**Table 1**

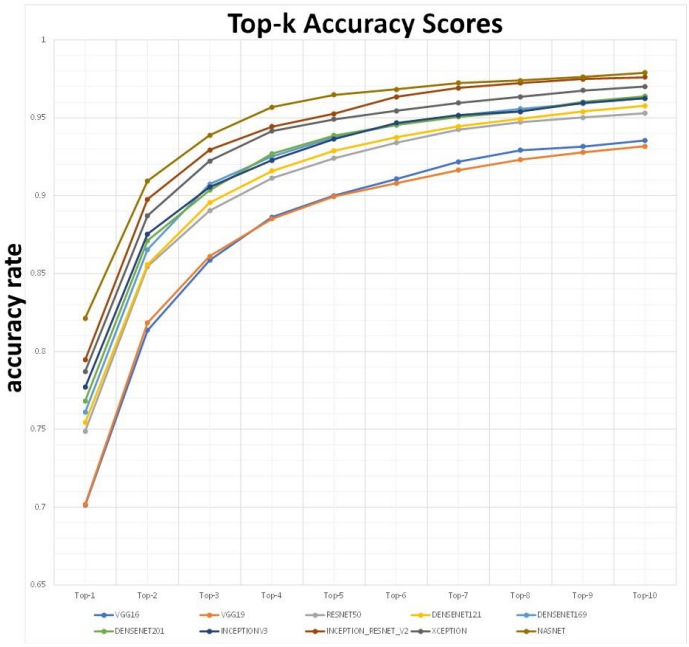


**Figure 10**

### 5.3 Discussion

The scores observed were analogous to the release date of the state-of-the-art classification model. NASNet model was observed to outperform all other models. The accuracy was observed to drop steeply beyond top-4 category for all models. NASNet and VGG models were the most memory consuming models, whereas DenseNet121 consumed significantly less memory than others.
All the models achieved accuracy scores better than an average human in the classification task.

## 6 Conclusions and Future Work

Although huge advances have been made in this area by using DNNs, they are computationally expensive to train, and do not provide reliable results to enable their use in a real-time application such as autonomous vehicles. Techniques such as the pruning of weights (to reduce the size) and domain transformation (Winograd or FFT, to gain speed) have been explored [11] [12], but none have been able to match the accuracy scores achieved by the state-of-the-art method. More research work on these areas may make it possible to use these techniques in a full-scale practical application.

## References


1. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (NIPS), 2012
2. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Proceedings of International Conference on Learning Representations (ICLR), 2015
3. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. CoRR, abs/1512.03385, 2015
4. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity Mappings in Deep Residual Networks. CoRR, abs/1603.05027, 2015
5. Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu and Kaiming He. Aggregated Residual Transformations for Deep Neural Networks. CoRR, abs/ 1611.05431, 2016
6. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9, June 2015
7. Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi.Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. CoRR abs/ 1602.07261
8. François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. CoRR abs/ 1610.02357
9. Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger. Densely Connected Convolutional Networks. CoRR. abs/1608.06993
10. Barret Zoph, Vijay Vasudevan, Jonathon Shlens, Quoc V. Le. Learning Transferable Architectures for Scalable Image Recognition. CoRR. abs/ 1707.07012
11. Andrew Lavin and Scott Gray. Fast algorithms for convolutional neural networks. CoRR, abs/1509.09308, 2015
12. Xingyu Liu, Yatish Turakhia. Pruning of Winograd and FFT Based Convolution Algorithm. Stanford edu, 2016. http://cs231n.stanford.edu/reports/2016/pdfs/117_Report.pdf


***Other Reference and Image sources:***


http://host.robots.ox.ac.uk/pascal/VOC/
http://image-net.org/challenges/LSVRC/2014/
https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803
https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8
https://towardsdatascience.com/understanding-and-visualizing-densenets-7f688092391a
https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852
https://medium.freecodecamp.org/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050
https://medium.com/zylapp/review-of-deep-learning-algorithms-for-image-classification-5fdbca4a05e2
https://hackernoon.com/evolution-of-image-recognition-and-object-detection-from-apes-to-machines-580ed4247f1e
https://www.technologyreview.com/s/530561/the-revolutionary-technique-that-quietly-changed-machine-vision-forever/
https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721
https://towardsdatascience.com/deep-learning-for-image-classification-why-its-challenging-where-we-ve-been-and-what-s-next-93b56948fcef