Image Classification using CNN

Study Project-1

Viraj Savaliya
ASU ID: 1217678787
Email: Viraj.Savaliya@asu.edu
Arizona State University, Tempe, AZ

Raunak
ASU ID: 1217240245
Email: rraunak@asu.edu
Arizona State University, Tempe, AZ

Nayankumar Patel
ASU ID: 1217420191
Email: npatel48@asu.edu
Arizona State University, Tempe, AZ

***Abstract*—State of Art image classification example used in Google photo application to filter the never tagged images or photos taken by camera with respect to individuals and places proves the use of convolutional neural network in image classification and computer vision applications. CNN's ability to take images' individual pixels as input for feature extraction and object detection makes it more robust compared to traditional methods which uses textures and shapes for classification. Use of such classification models is not limited to discussed application, but it used extensively in image based web search, medical diagnosis, autonomous driver assistance system etc.**

***Index Terms*—CNN, Neural Network**

I. Introduction

In Computer Vision - image classification is important aspect and has gained more importance in terms of its application. Traditional Learnable classifier used for image classification has difficulties in finding features from multiple images. On other hand by using CNN - this issue can be resolved and classification accuracy can also be improved by using layered neural networks and extracting more higher level of representation of layered images. In CNN, it has started with AlexNet followed by VGGNet, GoogleNet, ResNet, DenseNet etc. In this paper we are going to study how an image classification problem accuracy can be improved by adding an architectural unit in a CNN network. Image classification problems are interesting in many terms where in autonomous driving application model has to differentiate between different objects like Pedestrians, Bicycles, road, cars etc in real time scenarios and with minimum possible latency. In these application dynamic feature extraction make model more robust where it needs to differentiate different background pixels and object pixels in front of a self driving car. All these can be possible with Convoluted Neural Network with good computing capabilities in real time scenario.

II. Application

Conventional methods for image classification were using local handcrafted features and machine learning methods to perform image classification and recognition. These supervised machine learning methods required a large number of training samples which are labeled based on different classes. In real time scenarios, features which are handcrafted are not always optimal as they were extracted and expressed using a designed algorithm and individual experience. On other hand with further advancement in deep learning and evolved Neural Network Models used to perform feature extraction process through learning. Which outperforms the classical image recognition methods and infer what object they refer more accurately. Conventional labeled features and machine learning algorithm has human level accuracy for object detection problems where we have smaller labeled data-sets. But in realistic scenarios it is even difficult to train model for object detection and classification with millions of data-set and thousand categories.

Fig. 1. **Inter-Class Variability**

Fig. 2. **Intra-Class Variability**

Larger data-sets like ImageNet [1] consists 15 million images in 22,000 different categories. For this kind of data-set a simple machine learning model wont give required performance where classes are more complex and very fine grained. Here, even in different classes objects might look very similar and hard to distinguish - that means data-sets has more inter class variability - Fig. 1. As shown in the image, a single bird class has multiple different birds like flamingo, cock, pigeon, quail etc. which shows inter class variability. On other hand images from same class might also look very different where we have higher intra-class variability - Fig 2. As shown in the image, a fine grained orange class has even variability where images of orange looks very different.

Use of hidden layers for feature extraction and computing new features with the same input pixels is the unique aspect of

CNN for image classification. One of the such model proposed was AlexNet back in 2012 which had higher accuracy and 50% reduction in error rates compared to conventional classification algorithms. AlexNet has raised the bar and provided a base line model for object detection and classification. Later on with additional convolutions layers along with variation in filters VGGNet and GoogleNet has improved the accuracy. To achieve human level performance - a residual learning was introduced as part of ResNet which has reduced the errors rate in multiple folds compared to previous models. By digging deep into previous architectures, stacking multiple layers won't always improve and can actually make things worse. When these deep networks start converging - accuracy of the model starts saturating and degrades rapidly. This results in to a degradation problem which is resolved by adding identity shortcut connection which skip one or more layer connections. This is similar to a Long Term Short Memory where a forget gate controls the information flow to the next step. To resolve the degradation problem, if we analyse a worst case scenario where early layers of deeper model gives same accuracy and in case of rewarding scenario deeper model has more accuracy than shallow counter part. This problem is solved by using deep residual learning framework.

III. Architecture

Hierarchal Information present in images is extracted using convolution filters in CNNs [2]. This is done using these filters at different layers which aim to get different information at each stage. The lower layers try to find trivial information like edges or high frequencies and the upper layers detects the major information like face, text or shapes. By increasing the depth of the network, better performance can be achieved but the accuracy gets saturated after a point and it starts degrading rapidly. ResNet solves this problem by using identity-based skip connections and makes learning in deeper and stronger networks possible. The key element of making all this work is fusion of spatial and channel information of the image. SENets like SE-ResNet extends the existing ResNet by simply adding few additional parameters to the convolutional block where computation on these channels take place. Instead of weighing each channel equally, SENet performs context aware adaption of weights on each channel. In simple terms, it provides importance to each feature according to its relevance in the context of the application. The two major steps followed in SENets are Squeezing to get a global representation of all the features of the image and Excitation to provide each channel with its importance.

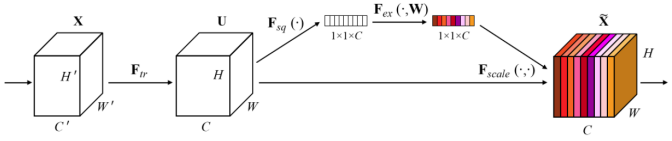

Fig. 3. **Squeeze and Excitation (SE) Block [3]**

SENets consists of a building block called SE Block which is represented as shown in Fig. 3. The input X which is of the size $H' \times W' \times C'$ consisting of information in different channels is transformed into a feature map U of size $H \times W \times C$. The transformation is done using a convolutional operator F_{tr} which can be either residual block or inception block depending on the network. This is done in order to represent the features in a suitable dimensionality before performing any operations on the feature channels. Then each channel undergoes squeeze operation which outputs a single numeric value using global average pooling. This numeric value is basically the channel descriptor aggregated across the spatial dimensions $H \times W$. The input to this squeeze operation in each channel is a collection of all the local descriptors which are expressive of the entire image. Such a global statistics of the network is embedded into a channel wise feature responses or channel wise statistics so that it can be used by all the layers. This is done to exploit the effect of channel dependency for contextual information outside the local regions of each channel. The statistic value is generated using the formula given by eqn. 1.

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j) \tag{1}$$

Global average pooling is the simplest way to aggregate the statistics. The squeeze operation is important as it plays an integral role in improving the accuracy and reducing the computational complexity of the network and is proven by research [3].

$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2 \delta(W_1 z)) \tag{2}$$

$$\mathrm{x}_c = F_{scale}(u_c, s_c) = s_c u_c \tag{3}$$

The second step introduced in SENets is the excitation operation. This operation can be shown by the eqn. 2. The excitation operation fully captures the channel-wise dependencies and is flexible and capable to learn a non-linear and non-mutually exclusive relationship between the channels. A simple gating mechanism using a sigmoid function is used for this purpose in the second fully connected layer. Here, sigmoid is used as the excitation operator as using other alternatives like ReLU or tanh might worsen the performance drastically as shown by research [3]. The first fully connected layer is followed by the ReLU function which adds the necessary non-linearity. The bottleneck with the two fully connected layers to limit the model complexity and to aid generalization is formed using a reduction ratio r. Careful selection of this reduction ratio is important in order to maintain a balance between the performance and computational complexity as there exists a trade-off between the two when the ratio is varied. The final output of the block is obtained by rescaling the transformed output after activation operation and can be shown by eqn. 3. The activations done on the channel produce channel weights which are adapted as per the input descriptor z. This helps to boost the discriminability between the features and provide a factor of importance to each feature channel.

$$\frac{2}{r} \sum_{s=1}^{s} N_s {C_s}^2 \tag{4}$$

The total number of additional parameters introduced by the SE block can be given by eqn. 4 [4]. Here r is reduction ratio, s is number of stages, C_s denotes dimension of output channels and N_s denotes repeated block number for stage s. However, the SE block is computationally lightweight, and the model complexity is only slightly affected by it's addition.

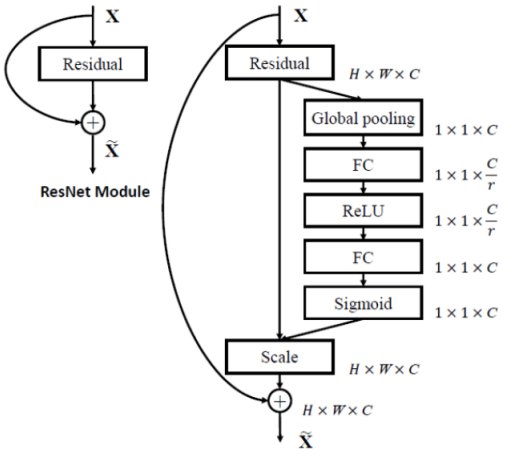

Fig. 4. **SE-ResNet Architecture [3]**

A SE network can be constructed by stacking collection of SE blocks or by simply replacing the original block at different stages in existing architectures. The role of the SE block is different at different layers in the network and it provides significant benefits at each stage and in turn boosts the overall performance of the model. The SE blocks are also quite robust in terms of their location while integrating them in existing architectures. The standard way of integrating the SE block is by placing it after the residual unit and before the summation with the identity branch in SE-ResNets. The Fig. 4 shows an example of how a Residual network when modified using SE block is instantiated in SE-ResNet model. Introducing SE blocks allows the SE-ResNet-50 to obtain the accuracy levels of ResNet-101 model with only a slight increase on computational complexity than ResNet-50 but almost half the complexity than ResNet-101.

For Implementing and training SENets, resources like TensorFlow, Keras or PyTorch and GPUs having atleast 12GB standard memory space and 3584 CUDA cores are preferred. The experiment performed by the authors [3] for SE-ResNet was using 8 NVIDIA Titan X GPUs. However, the training can be done using less GPUs or also using CPUs but will have significant increase in the total computation time for the model. Also, the hyperparameters like Reduction ratio, initial learning rate, batch size, weight decay, momentum, epochs, optimzation algorithm and kernel width are needed to be tuned before training for implementing SE-ResNet model. Poor tuning of these parameters can lead to worse performance by the network. The authors of SE-ResNet [3] have used synchronous SGD for optimization with momentum 0.9, batch size of 256, initial learning rate of 0.1 with decrease of factor 10 when loss plateaus, weight decay of 0.0001, reduction ratio as 16 and kernel width of 224 x 224 pixels.

IV. Comparison

Deeper convolutional neural networks have achieved major breakthroughs in the image classification arena. But they lead to a degradation problem as number of layers increases. With the network depth increasing, the accuracy starts getting saturated. This degradation problem is solved by using a deep residual learning framework in ResNet. Which is one of the basis architecture for SENet and is capable of solving the image classification problem with a respectable accuracy.

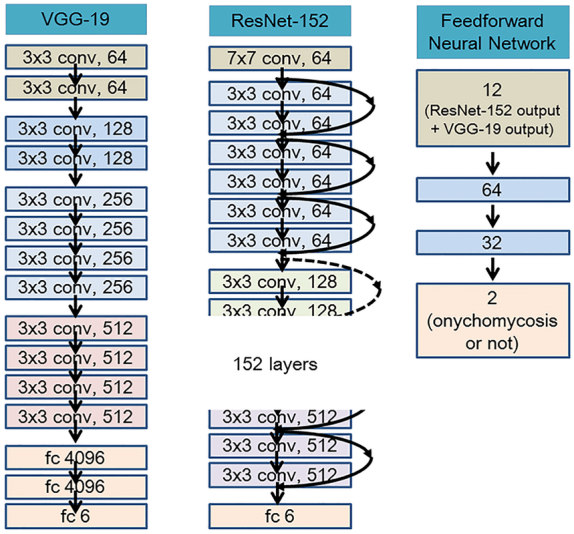

Fig. 5. **VGG-net and Resnet Neural Network**

A. Architecture

ResNet architecture is inspired by VGG-Net. 3x3 filters are used in the convolutional layers and they adhere to two design rules [5],

1. Layers have same number of filters for same output feature map size.

2. To preserve the time complexity layer, the number of filters are doubled as the feature map size is halved.

Downsampling is performed directly by the convolutional layers with a stride of 2 and a network ending with global average pooling layer and 1000-way fully connected layer with softmax is created.

Taking the above network as the basis, shortcut connections are inserted into the network, which converts the whole network into its residual version. These identity shortcuts are directly used when the input and output are of same dimensions. when the dimensions are different extra zero padding is used to match the dimensions.

In case of SENet it uses squeeze and excitation blocks and each SE block makes use of global average pooling operation in the squeeze phase and two fully connected layer in the excitation phase of the network and these SE blocks can be used in any existing architectures to form a SENet version [3].

B. Implementation

ResNet is implemented on the imagenet dataset. Image is resized and cropped to 224x224 and is randomly sampled

Layer Name	Output Size	ResNet-18
conv1	112 × 112 × 64	7 × 7, 64, stride 2
conv2_x	56 × 56 × 64	3 × 3 max pool, stride 2 $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$
conv3_x	28 × 28 × 128	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$
conv4_x	14 × 14 × 256	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$
conv5_x	7 × 7 × 512	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$
average pool	1 × 1 × 512	7 × 7 average pool
fully connected	1000	512 × 1000 fully connections
softmax	1000	

Fig. 6. **ResNet architecture**

from the image or its horizontal flip with the subtracted per pixel mean. Batch normalization is used after each convolution and before activation. weights are initialized and trained from scratch. Stochastic gradient descent with a mini batch size of 256 is used. Starting learning rate is set to 0.1 and each time it is divided by 10 when the error plateaus and total iterations for the training of the model is taken as 600000. Weight decay of 0.0001 and a momentum of 0.9 is used [5]. A comparison between the VGG-19 and ResNet model is shown in Fig. 5.

For testing, standard 10-crop testing is adopted. Various layers with their output sizes and the ResNet-18 architecture is depicted in the Fig. 6.

SENet is also trained on imagenet dataset, which comprises of 1.28 million images and 50k validation images from different classes. Images are resized and cropped to 224x224 here too and takes the input of 224x224 pixels size. Each input is normalized through mean RGB-channel subtraction. Stochastic gradient descent is used with momentum of 0.9 and a mini batch size of 1024. Starting learning rate is set to 0.6 and is decreased by a factor of 10 for every 30 epochs. Models are trained for 100 epochs from scratch [3].

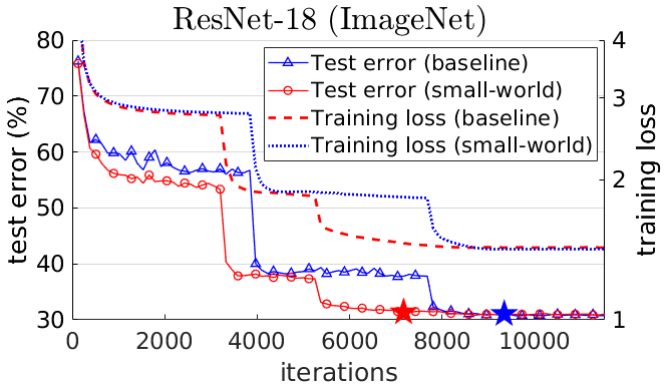

Fig. 7. **ResNet performance**

C. Evaluation

ResNet was evaluated on the Imagenet dataset [1]. 152-layer residual net is the deepest network ever presented on Imagenet. It has lower complexity than VGG-net. It has 3.57 percent top-5 error on the imagenet test-set and it won 1st place in the ILSVRC 2015 classification competition. It also won 1st places in ImageNet detection [5], ImageNet localization, COCO detection and COCO segmentation in ILSVRC and COCO 2015 competitions. This strong evidence shows that it is capable of performing the image classification task with great ease and decent accuracy as seen in Fig. 7 . ResNet benchmark performance with respect to various existing architecture is depicted in Fig. 8.

Since ResNet is a variation of deep convolutional neural network with shortcut residual connections added to mitigate the degradation problem with the deeper convolutional network, so deeper convolutional networks can be constructed with the help of ResNet to solve the image classification problem.

method	top-5 err. (**test**)
VGG [41] (ILSVRC'14)	7.32
GoogLeNet [44] (ILSVRC'14)	6.66
VGG [41] (v5)	6.8
PReLU-net [13]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	**3.57**

Fig. 8. **ResNet benchmark**

V. Conclusion

In this study project, we solve the image classification problem using convolutional neural network. Deep convolutional networks extract the low/mid/high level features from the images and classifiers in an end to end procedure and these features can be increased as we increase the number of stacked layers leading to an efficient image classification technique. A recent CNN technique, SENet and its architecture is described to solve the image classsification problem. SENet is composed of squeeze and excitation blocks, which performs dynamic channel-wise feature recalibration to improve the representational power of the CNN network. Squeeze and excitation blocks can be inserted in any existing CNN architecture to form SENet to improve its image classification accuracy.

We further, compared the SENet model with the ResNet model, and listed out the differences in architecture, implementation and evaluation of the model. ResNet eliminates the degradation problem in deeper convolutional network and enables the formation of deeper ResNet models to achieve better accuracy than the existing VGG-net, inception and GoogleNet models as shown in Fig. 8.

ResNet is the basic block of the SENet model. SENet is able to model channel-wise feature dependencies which the existing architectures such as, ResNet failed to do so. Thus, a wide range of experiments reveal the effectiveness of SENets, which achieves state of the art across multiple tasks and datasets to solve the image classification problem.

CONTRIBUTION

Name	Sections
Nayankumar Patel	Abstract, Introduction, Application
Viraj Savaliya	Architecture
Raunak	Comparison, Conclusion

REFERENCES

[1] "Imagenet dataset." [Online]. Available: http://www.image-net.org/

[2] P.-L. Pröve, "Squeeze-and-excitation networks," Oct 2017. [Online]. Available: https://towardsdatascience.com/squeeze-and-excitation-networks-9ef5e71eacd7

[3] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[4] S.-H. Tsang, "Review: Senet-squeeze-and-excitation network, winner of ilsvrc 2017 (image classification)," Oct 2019. [Online]. Available: https://towardsdatascience.com/

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.