# Training AlexNet and ResNet on Cifar-10

University of Washington
Electrical and Computer Engineering
Yetao Chen, Yilin Wang, Sheng-hao Chen

## 1. Abstract

In our project, we choose to implement AlexNet and ResNet from scratch in Pytorch, and train these 2 networks on Cifar-10 dataset.
We plan to tune the parameters in the training process, in order to get the best performance. We succeeded in getting above 85% testing accuracy for AlexNet, and above 90% testing accuracy for ResNet.

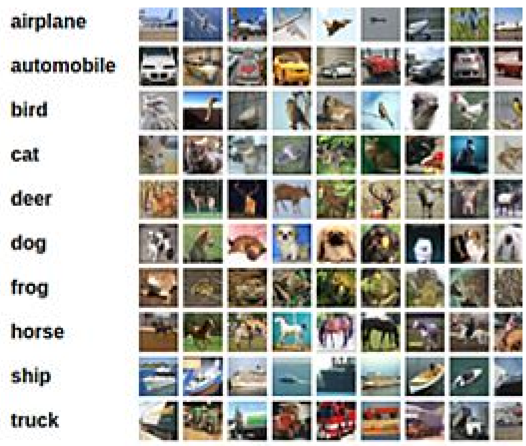
## 2. Introduction

Computer vision has become a popular realm for recent years due to the performance of hardware. It makes a great step, which combines machine learning with multiple layers to enhance accuracy in predictions. In 1998, the mother of CNN, LeNET, proposed an unprecedented architecture to process images with convolution layers and fully-connected layers to classify the images. In 2012, AlexNet, designed by Alex Krizhevsky in collaboration with Ilya Sutskever and Geoffrey Hinton, won the champion on ImageNet LSVRC and started a new age of CNN. After that, VGGNet modified parts of parameters on AlexNet and added more layers to obtain more amazing predictions accuracy. ResNet, based on VGGNet19, utilizes skip connections, or shortcuts to jump over some layers. Smaller feature map sizes, but more feature maps that the way they deployed keeps the complexity and forms residual learning to reach higher accuracy.

In this project, we implemented a deep convolutional neural network (CNN) for image classification. We used a customized dataset with 10 classes (i.e. face, airplane, dog, car, tree and so forth), and the dataset contains a thousand 32 X 32 color images per class. There are 60,000 images in total for 10 classes, that is 6000 images/class. There is a training set of 50,000 images and a test set of 10,000 images. We classified the images based on the content and then acquired our predictions to their labels. Our goal is to get testing accuracy for AlexNet and ResNet. In addition, we also try to modify specific parameters to observe the differences like learning, batch sizes and input image size.

## 3. Review and Prepare CIFAR10 Dataset

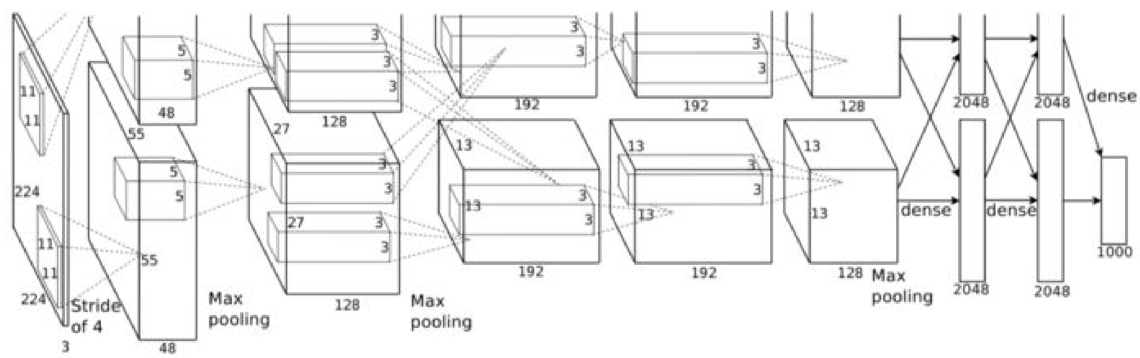
As a Computer Vision practitioner, CIFAR10 is commonly the first dataset to experience. This dataset contains 60000 32x32 color images uniformly distributed to 10 classes. It is divided into five training batches and one test batch, each with 10000 images.




4. **Review Modern CNN Architectures**

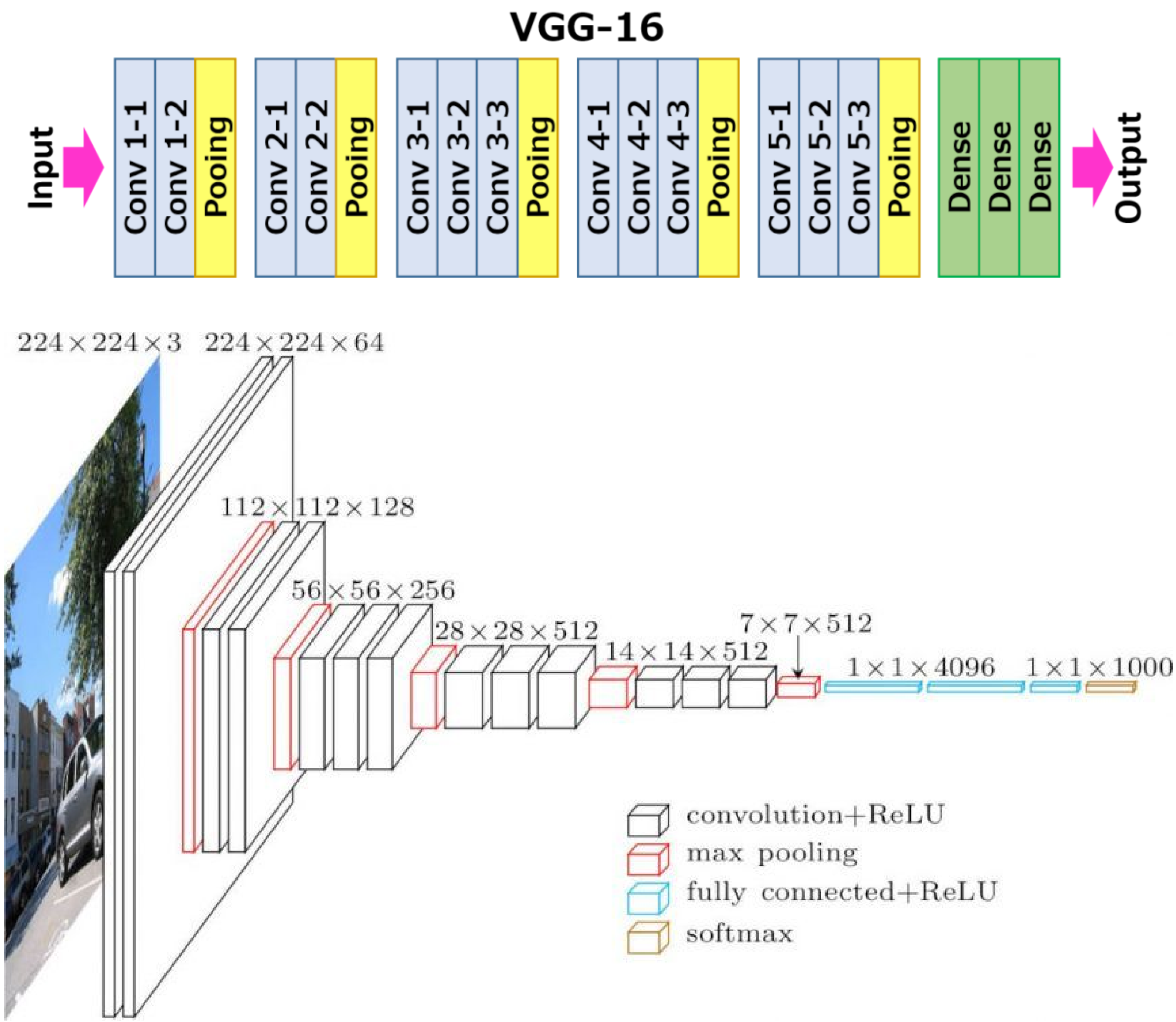
   **4.1 AlexNet network**

   The architecture consists of eight layers: five convolutional layers and three fully-connected layers. AlexNet using REctified Linear Units (ReLU) is able to reach a 25% error on the CIFAR-10 dataset six times faster than a CNN using tanh. AlexNet has 60 million parameters, a major issue in terms of overfitting, which we try to resolve in later parts.



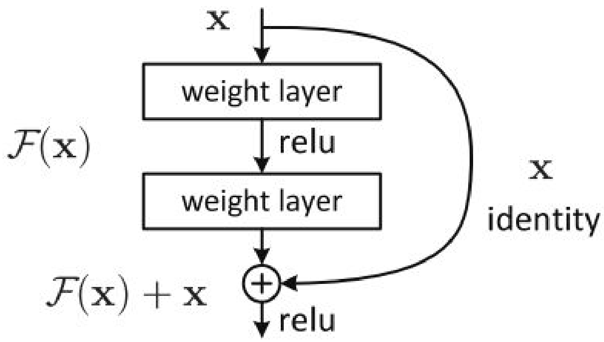

**4.2 VGG network**

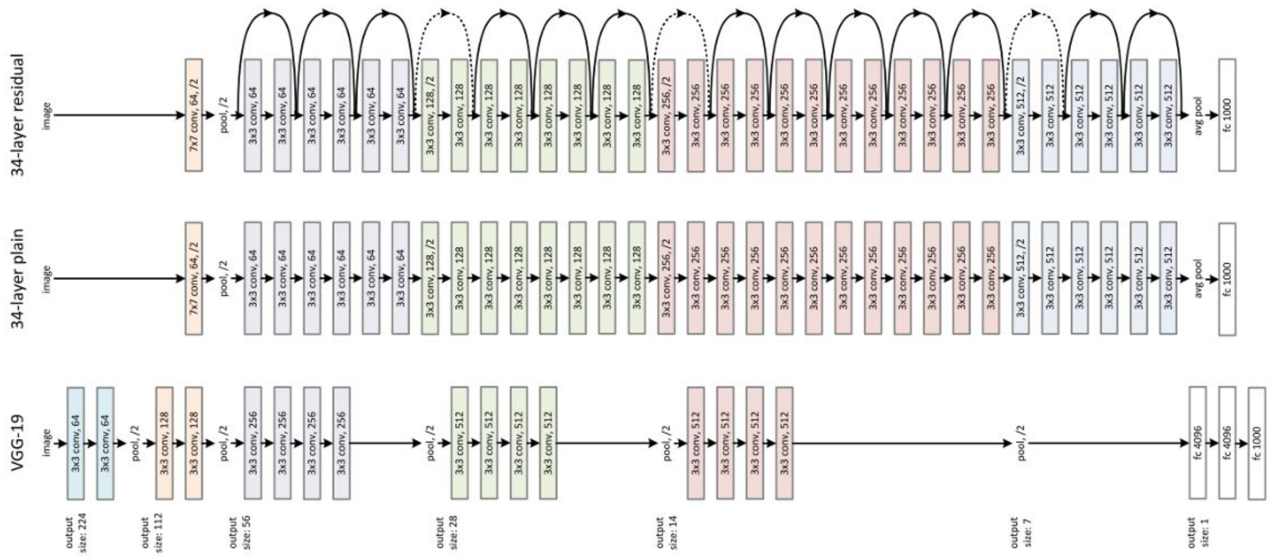
VGGNet is the 1st runner-up in ILSVRC 2014 in the classification task. It is the first network that uses blocks as unit components. Each block consists of a series of convolutional layers, followed by a max pooling layer for spatial downsampling. VGG16 makes the improvement over AlexNet by replacing large Kernel-sized filters(11 and 5 in the first and second convolutional layer, respectively) with multiple 3x3 kernel-sized filters one after another. VGG16 has 138 million parameters, could reach 8.8% top-5 error rate. Later on, inherited from the ideas, VGGNet is customized to use 5x5, 7x7 kernels for specific situations.




### 4.3 ResNet network

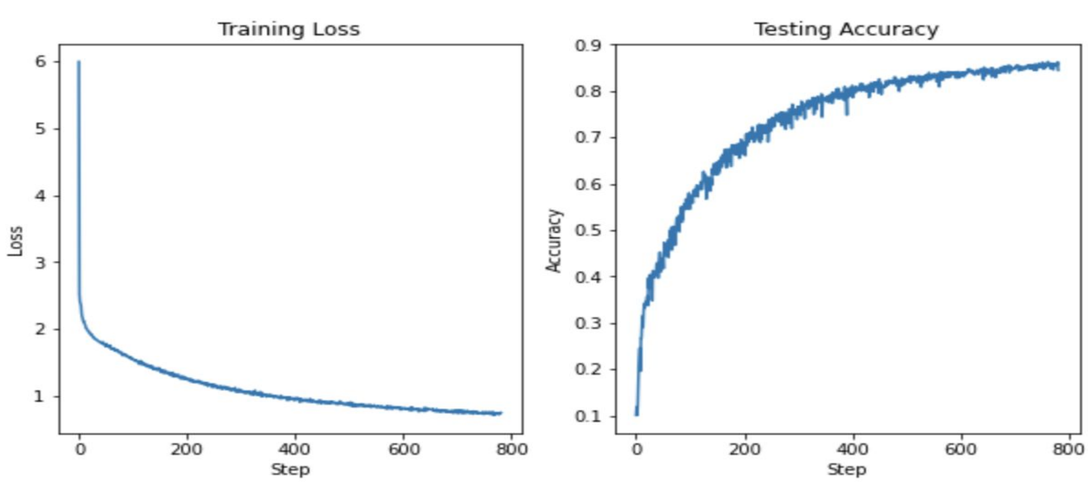
ResNet is the most ubiquitous network that can be applied in many applications and its efficiency is clearly proved after the winning at ILSVRC 2015. ResNet follows VGG's 3x3 convolutional layer design and it introduces the residual block with the additional Identity mapping by Shortcuts. The residual blocks not only help increase the function complexity but also reduce the training errors. The core idea of ResNet is introducing a so-called "identity shortcut connection" that skips one or more layers, as shown in the following figure:




## 5. Experiments and Results

***AlexNet:***



To reduce overfitting, one method is data augmentation, we used label-preserving transformation to make data more varied. Specifically, we generated image rotation and horizontal reflections.

There are numerous options for optimizer e.g. Gradient Descent, Stochastic Gradient Descent, Adam ... to determine how to change the value to minimize the loss function for every parameter. In this work, we utilize SGD optimizer.

At first, when batch_size = 32, learning rate = 0.001, only 50 epochs are trained,  we got testing loss = 0.416, and testing accuracy = 0.583, which is obviously too low compared with  normal level.

Then in another experiment, when batch_size = 64, learning rate = 0.01,  epochs = 100. We could get testing Loss = 0.424 and testing accuracy =  0.862.

***Tuning***

Image input size has a great impact on testing accuracy. In the beginning, we tried to use a smaller input image which is 64x64. We found that the overfitting status is very obvious after 20 epochs and testing accuracy only can reach around 75%(Fig. A). Compared to the same model but different program we mentioned above, it is fairly low, not as we expected.

(lr=0.01, momentum=0.9, momentum=0.9 batch_size=64, epoch=100, Training Loss = 0.423)

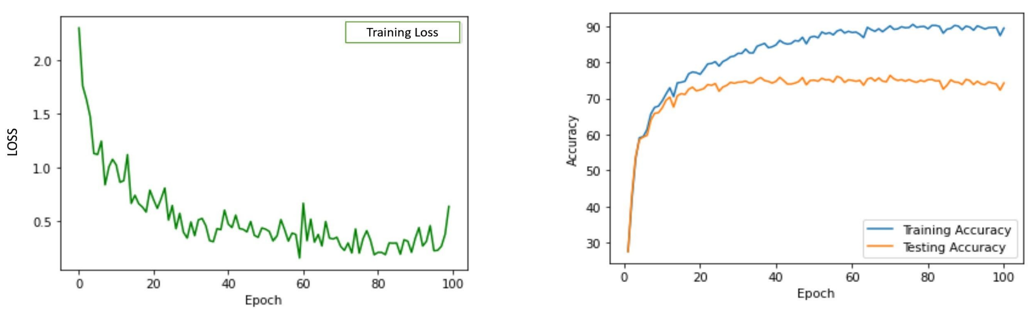



Fig. A

We followed the originate design of AlexNet and resized to 224x224. At this time, we set the learning rate to be 0,001 and batch size to be 128.(We've already found that if the learning rate is 0.001, we can get best testing accuracy. Too small learning rate leads to updates stucking in local minimums, but too large learning rate leads to divergent updates.) The testing accuracy is up to about 82% through 50 epochs(Fig. B).

(lr=0.001, momentum=0.9, batch_size=128, epoch=50, Training Loss = 0.344)

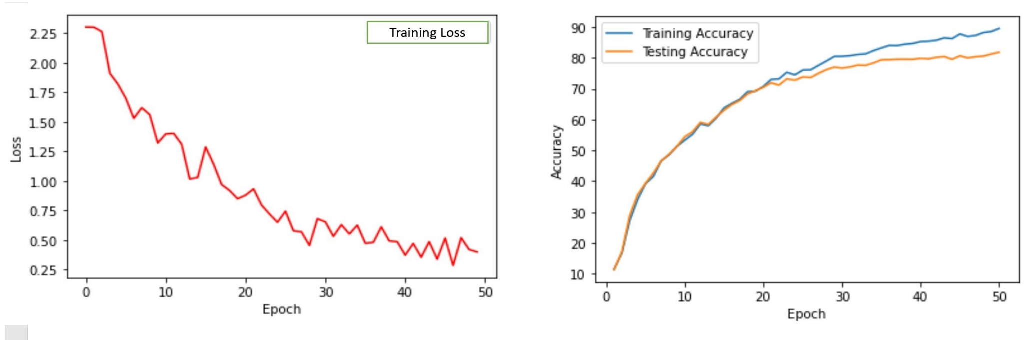
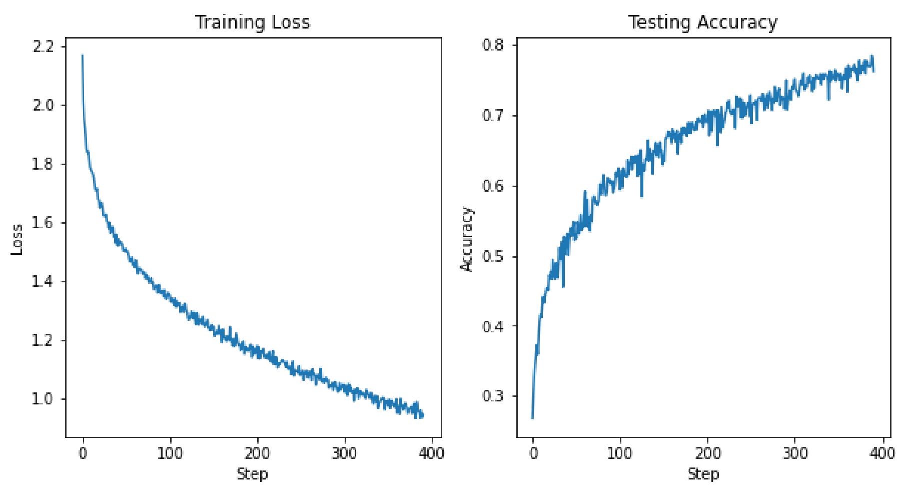



Fig. B

***ResNet:***

Using cross entropy loss, SGD, with data augmentation.

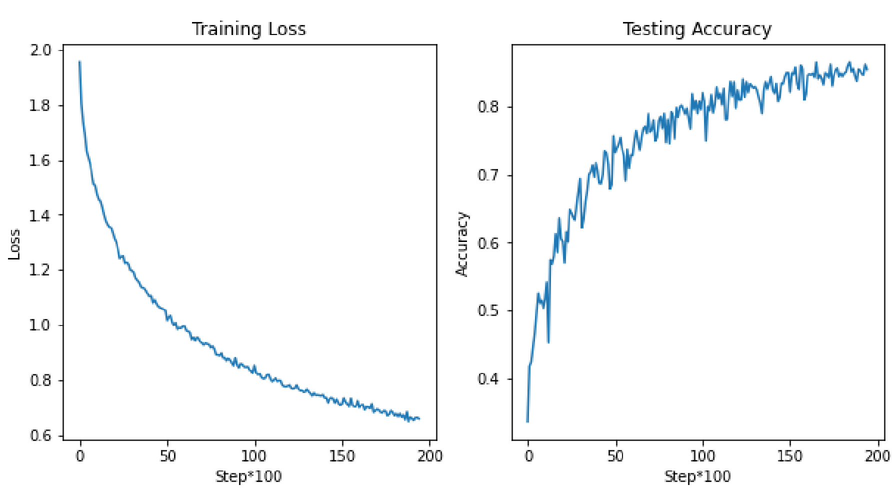
1. batch_size=64, learning rate=0.001, epochs=50




**Testing accuracy = 0.762**

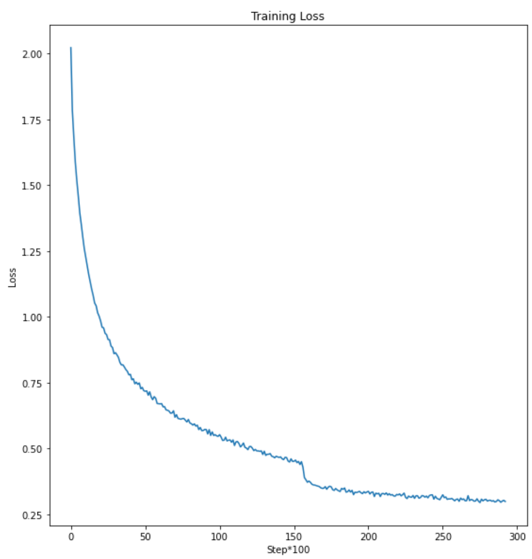
Because of the low learning rate, small batch size and few epochs, the testing accuracy is low and far from convergence. The training process is very slow, so we have two choices: 1. More epochs i.e. takes more time. 2. Increase the learning rate and batch size to make the training faster.

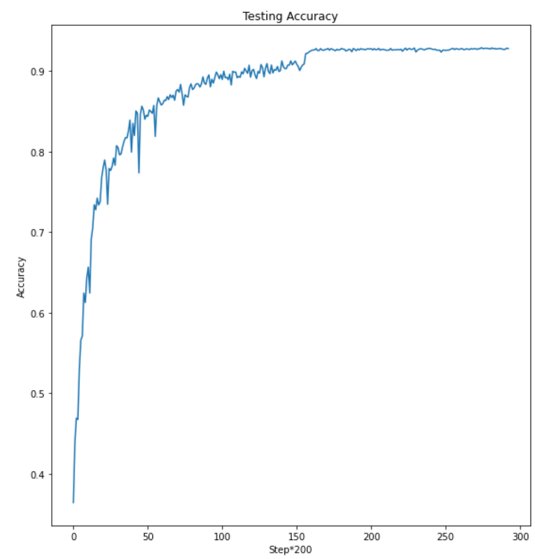
2. batch_size=128, learning rate=0.01, epochs=50



**Testing accuracy = 0.854**
We increased our learning rate and batch size to make the training faster. However, the result is still not as good as we expected. It is still not convergent. We can train it for more epochs, but if the learning rate is too high, the loss will fluctuate forever and never get to the minimum point. If the learning rate is too low, the training process may get very long and the loss could get stuck into the local minimum.
In order to make our training more efficient, we decided to schedule our learning rate: high in the first several epochs and decrease it later.

3. batch size=128, epochs = 150, learning rate schedule: 0.1 when epoch < 80; 0.01 when 80 <= epoch < 130; 0.001 when 130 <= epoch <= 150.




**Testing accuracy = 0.929.**
When we decreased our learning for the first time (at the epoch of 80), the performance got significantly better, and got convergent later. This means that scheduling the learning rate really made a difference on optimizing the performance.

## 6. Conclusion

As a result, we achieved our goal to get the best testing accuracy 86.2% on AlexNet, and 92.9% on ResNet. How to efficiently train the network with limited resources is the most important challenge we faced. Tradeoffs between batch size, epochs and learning rate, and the reasonable schedule of learning rates can make the network get the best performance more efficiently.

## 7. Reference

1. Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems.

2. Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun Microsoft Research (2015). Deep Residual Learning for Image Recognition. Microsoft Research