ImageNet Classification with Deep Convolutional Neural Networks
Large-scale Video Classification with Convolutional Neural Networks

Zhuoyan Xu

University of Wisconsin-Madison

March 12, 2019

Structure

Overview

- Both papers are the methodology and application of Convolutional neural network. The first one is on the imageNet dataset. The second one is on the YouTube vedio dataset.
- The two paper used samilar architecture and tricks of CNN, the video classification is more complicated.
- In this report, we will briefly cover the architecture they use and the performance of their model.

Structure

Basic part

- Classify the 1.2 million high-resolution images into the 1000 different classes.
- The Network contains a number of new and unusual features which improve its performance and reduce its training time.
- Several effective techniques for preventing overtting
- Networks size is limited mainly by the amount of memory available on current GPUs and by the amount of training time that they are willing to tolerate. The result can be improved simply by waiting for faster GPUs and bigger datasets to become available.

dataset

- The dataset is a subset of imageNet. In all, there are roughly 1.2 million training images, 50,000 validation images, and 150,000 testing images.
- Given a rectangular image, they first rescaled the image such that the shorter side was of length 256, and then cropped out the central 256×256 patch from the resulting image. Didn't pre-process the images in any other way, except for subtracting the mean activity over the training set from each pixel. Thus, they trained our network on the (centered) raw RGB values of the pixels.

ReLu Nonlinearity

- Why nonlinearity? Image has lots of nonlinear elements, transition between pixes adjacent pixes is often nonlinear. The borders, different colors are nonlinear. However, When create network, it might create something linear.
- Saturation occurs when the hidden units of a neural network (NN) predominantly output values close to the asymptotic ends of the activation function range. Saturation reduces the NN to a binary state, thus limiting the overall information capacity of the NN. Saturated units make gradient descent learning slow and inefficient due to small derivative values near the asymptotes[A. Rakitianskaia. *Measuring Saturation in Neural Networks*].
- Deep convolutional neural networks with ReLUs units train several times faster than that with *tanh, sigmoid* units.

Training on multiple GPUs

- The parallelization scheme essentially puts half of the kernels (or neurons) on each GPU, with one additional trick: the GPUs communicate only in certain layers.
- This scheme reduces top-1 and top-5 error rates by 1.7% and 1.2%.

Local response normalization

Denoting by $a^i_{x,y}$ the activity of a neuron computed by applying kernel i at position (x,y) and then applying the ReLU nonlinearity, the response-normalized activity $b^i_{x,y}$:

$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} \left(a^j_{x,y}\right)^2 \right)^{\beta}$$

where the sum runs over n adjacent kernel maps at the same spatial position, and N is the total number of kernels in the layer.

Response normalization reduces our top-1 and top-5 error rates by 1.4% and 1.2%, respectively.

Overlapping pooling

- A pooling layer can be thought of as consisting of a grid of pooling units spaced s pixels apart, each summarizing a neighborhood of size z×z centered at the location of the pooling unit. In this paper, they choose s=2,z=3.
- This scheme reduces the top-1 and top-5 error rates by 0.4% and 0.3%, respectively, as compared with the non-overlapping scheme s = 2,z = 2.

Overall architecture

- The net contains eight layers with weights; the first five are convolutional and the remaining three are fully connected. The output of the last fully-connected layer is fed to a 1000-way softmax which produces a distribution over the 1000 class labels.
- The kernels of the second, fourth, and fifth convolutional layers are connected only to those kernel maps in the previous layer which reside on the same GPU.
- Response-normalization layers follow the first and second convolutional layers.
- Max-pooling layers, follow both response-normalization layers as well as the fifth convolutional layer.
- The ReLU is applied to the output of every convolutional and fully-connected layer.

Architecture

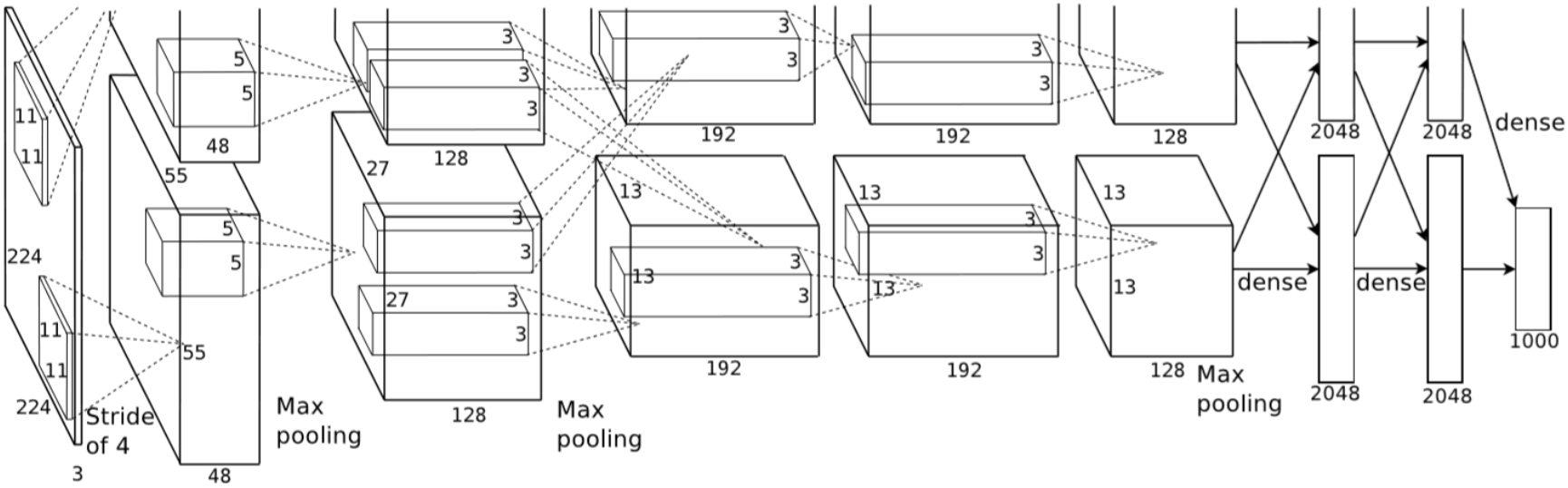


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

Figure: Architecture

Data Augmentation

- There are 2 transformations, both of them cost little computation, so the transformed images do not need to be stored on disk.
- The first one is generating image translations and horizontal reflections. Trade off: The resulting training examples are highly interdependent. Without this scheme,the model is overfitting, which lead to much smaller networks.
- The second is altering the intensities of the RGB channels in training images. Specically, perform PCA on the set of RGB pixel values throughout the ImageNet training set.

Data Augmentation

For each RGB image pixel $I_{xy} = [I_{xy}^R, I_{xy}^G, I_{xy}^B]^T$, add the following quantity:

$$[\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3] [\alpha_1\lambda_1, \alpha_2\lambda_2, \alpha_3\lambda_3]^T$$
$$\alpha_i \ i.i.d \ \sim \ N(0, 0.1)$$

where p_i and λ_i are i-th eigenvector and eigenvalue of the 3×3 covariance matrix of RGB pixel values. Each α_i is drawn only once for all the pixels of a particular training image until that image is used for training again.

This scheme approximately captures an important property of natural images, i.e.that object identity is invariant to changes in the intensity and color of the illumination. This scheme reduces the top-1 error rate by over 1%.

Dropout

Use dropout in the first two fully-connected layers of architecture. Without dropout, the network exhibits substantial overfitting. Dropout roughly doubles the number of iterations required to converge.

SGD

Using SGD with a batch size of 128 examples, momentum of 0.9, and weight decay of 0.0005.This small amount of weight decay was important for the model to learn.
The update rule for weight is:

$$v_{i+1} := 0.9 \cdot v_i - 0.0005 \cdot \epsilon \cdot w_i - \epsilon \cdot \left\langle \left. \frac{\partial L}{\partial w} \right|_{w_i} \right\rangle_{D_i}$$

$$w_{i+1} := w_i + v_{i+1}$$

where i is the iteration index, v is the momentum variable, ϵ is the learning rate. D_i is the i-th minibatch.

Result

Model	Top-1	Top-5
Sparse coding [2]	*47.1%*	*28.2%*
SIFT + FVs [24]	*45.7%*	*25.7%*
CNN	**37.5%**	**17.0%**

Table 1: Comparison of results on ILSVRC-2010 test set. In *italics* are best results achieved by others.

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SIFT + FVs [7]	—	—	*26.2%*
1 CNN	40.7%	18.2%	—
5 CNNs	38.1%	16.4%	**16.4%**
1 CNN*	39.0%	16.6%	—
7 CNNs*	36.7%	15.4%	**15.3%**

Table 2: Comparison of error rates on ILSVRC-2012 validation and test sets. In *italics* are best results achieved by others. Models with an asterisk* were "pre-trained" to classify the entire ImageNet 2011 Fall release. See Section 6 for details.

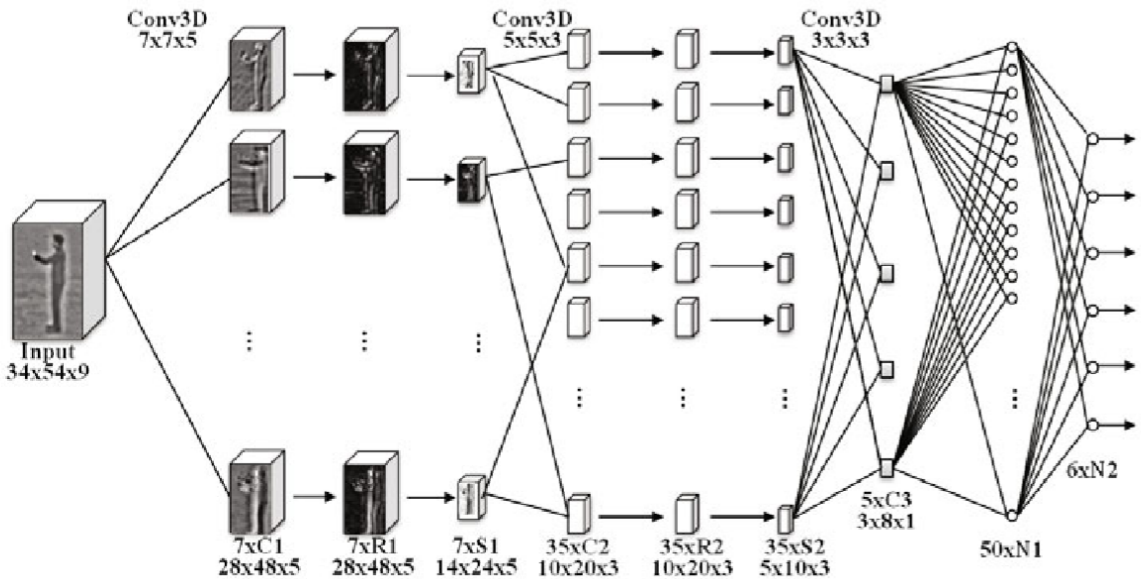
Structure

Background

- Use multiple approaches for extending the connectivity of a CNN in time domain to take advantage of local spatio-temporal information.
- Using a new dataset of 1 million YouTube videos belonging to 487 classes(which called Sports-1M).
- Evaluating multiple CNN architectures that each take a different approach to combining information across the time domain.
- Millions of parameters(or more) when extending the connectivity of the architecture in time. Mitigate this issue by modifying the architecture to contain two separate streams of processing: a context stream and a fovea stream.

Related work

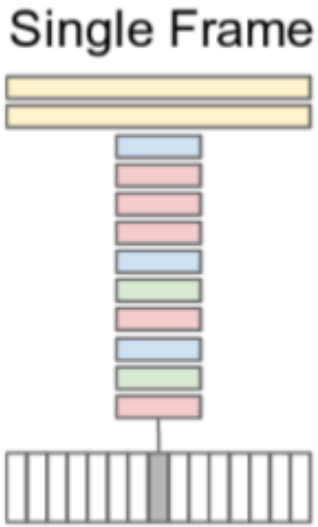
Extend an image CNN to video domains by treating space and time as equivalent dimensions of the input and perform convolutions in both time and space. [M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. *Sequential deep learning for human action recognition*].
Replace 2-D convolutions with 3-D:



Architecture

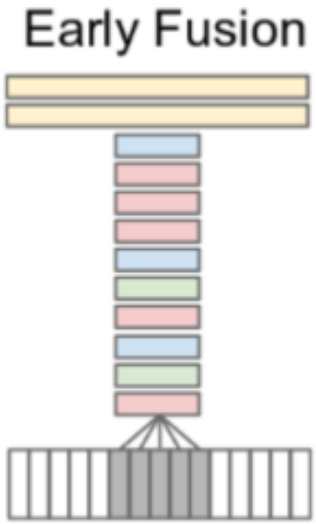
- Treat every video as a bag of short, fixed-sized clips. Since each clip contains several contiguous frames in time, we can extend the connectivity of the network in time dimension to learn spatio-temporal features.
- There are multiple options for how to extended connectivity (Time information fusion):Early Fusion,Late Fusion and Slow Fusion.
- Finally, a multiresolution architecture for addressing the computational efficiency.

Single-frame



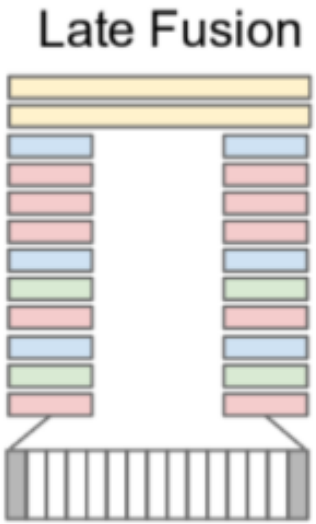
The full architecture is C(96,11,3)-N-P-C(256,5,1)-N-P-C(384,3,1)C(384,3,1)-C(256,3,1)-P-FC(4096)-FC(4096), where C(d,f,s) indicates a convolutional layer with d filters of spatial size f$\times$f,applied to the input with stride s. FC(n) is a fully connected layer with n nodes.

Early Fusion



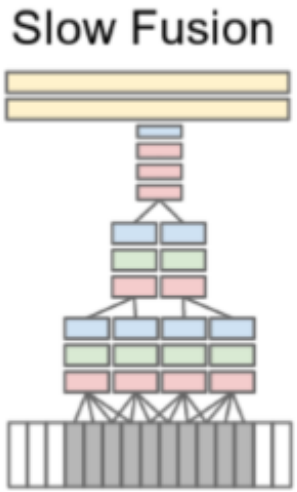
The Early Fusion extension combines temporal information across directly on pixel level. This is implemented by modifying the filters on the first convolutional layer in the single-frame model by extending them to be of size $11 \times 11 \times 3 \times T$ pixels, where T is some temporal extent. The early and direct connectivity to pixel data allows the network to precisely detect local motion direction and speed.

Late Fusion



The Late Fusion model places two separate single-frame networks with shared parameters a distance of 15 frames apart and then merges the two streams in the first fully connected layer. Therefore, neither single-frame tower alone can detect any motion, but the first fully connected layer can compute global motion characteristics by comparing outputs of both towers.

Slow Fusion



The Slow Fusion model is a balanced mix between the two approaches that slowly fuses temporal information throughout the network such that higher layers get access to progressively more global information in both spatial and temporal dimensions.

Multiresolution CNNs

- Since CNNs normally take on orders of weeks to train on large-scale datasets even on the fastest available GPUs, reduce computation complexity is more important. This paper focus on changing architectures to reduce running time.
- **Fovea and context streams** A 178×178 frame video clip forms an input to the network. The context stream receives the downsampled frames at half the original spatial resolution (89×89 pixels), while the fovea stream receives the center 89×89 region at the original resolution.

Multiresolution

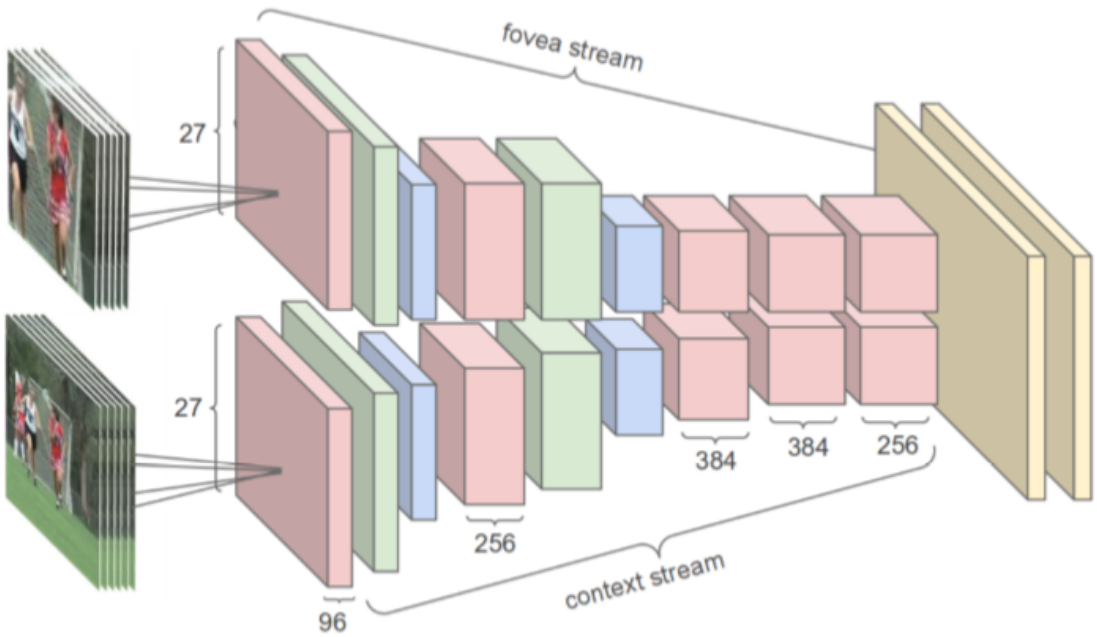


Figure 2: Multiresolution CNN architecture. Input frames are fed into two separate streams of processing: a *context stream* that models low-resolution image and a *fovea stream* that processes high-resolution center crop. Both streams consist of alternating convolution (red), normalization (green) and pooling (blue) layers. Both streams converge to two fully connected layers (yellow).

Result

Model	Clip Hit@1	Video Hit@1	Video Hit@5
Feature Histograms + Neural Net	-	55.3	-
Single-Frame	41.1	59.3	77.7
Single-Frame + Multires	**42.4**	**60.0**	**78.5**
Single-Frame Fovea Only	30.0	49.9	72.8
Single-Frame Context Only	38.1	56.0	77.2
Early Fusion	38.9	57.7	76.8
Late Fusion	40.7	59.3	78.7
Slow Fusion	**41.9**	**60.9**	**80.2**
CNN Average (Single+Early+Late+Slow)	41.4	63.9	82.4

Table 1: Results on the 200,000 videos of the Sports-1M test set. Hit@k values indicate the fraction of test samples that contained at least one of the ground truth labels in the top k predictions.

Contributions of motions

Figure 5: Examples that illustrate qualitative differences between single-frame network and Slow Fusion (motion-aware) network in the same color scheme as Figure 4. A few classes are easier to disambiguate with motion information (left three).

- Choose the Slow Fusion network as a representative motion-aware network because it performs best.
- Observe that motion aware networks are more likely to underperform when there is camera motion present. Hypothesis: CNNs struggle to learn complete invariance across all possible angles and speeds of camera translation and zoom.

Conclusion

- CNN architectures are capable of learning powerful features from weakly-labeled data that far surpass feature based methods in performance and that these benefits are surprisingly robust to details of the connectivity of the architectures in time.
- Slow Fusion model consistently performs better than the early and late fusion alternatives. However, a single-frame model already displays very strong performance, which suggest that local motion may not be critically important, even for a dynamic dataset such as Sports.
- An alternative theory is that more careful treatment of camera motion may be necessary, but this requires significant changes to a CNN architecture that can be left for future work.
- Mixed-resolution architectures consisting **context** and **fovea** stream can speed up CNNs efficiently without reducing accuracy.