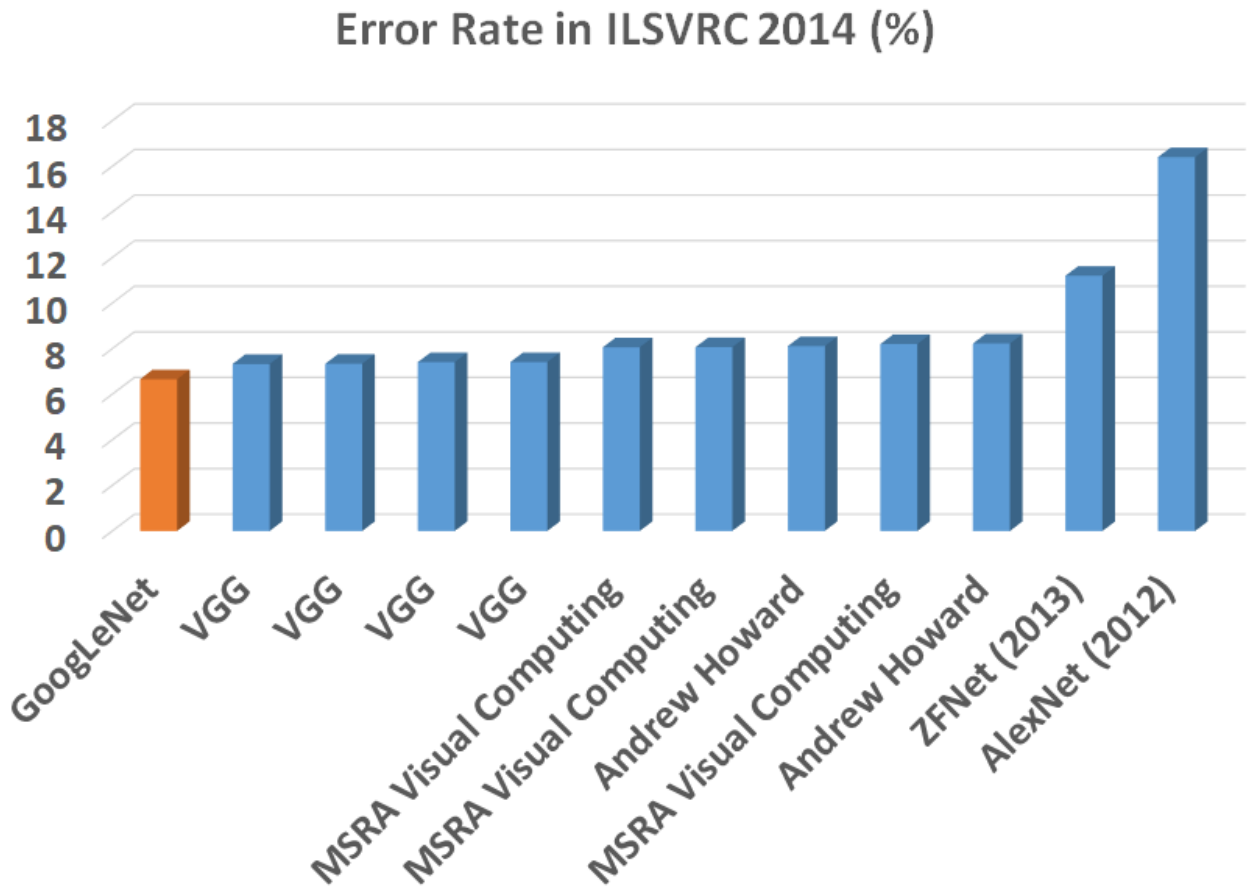
Review: GoogLeNet (Inception v1)— Winner of ILSVRC 2014 (Image Classification)

Sik-Ho Tsang
Aug 24, 2018 · 7 min read

In this story, **GoogLeNet [1]** is reviewed, which is the winner of the **ILSVRC (ImageNet Large Scale Visual Recognition Competition) 2014**, an image classification competition, which has significant improvement over ZFNet (The winner in 2013) [2] and AlexNet (The winner in 2012) [3], and has relatively lower error rate compared with the VGGNet (1st runner-up in 2014) [4].

From the name "**GoogLe**Net", we've already known that it is from Google. And "Goog**LeNet**" also contains the word "LeNet" for paying tribute to Prof. Yan LeCun's LeNet [5]. This is a **2015 CVPR** paper with **about 9000 citations** when I was writing this story. (Sik-Ho Tsang @ Medium)

It is also called **Inception v1** as there are v2, v3 and v4 later on.



ILSVRC 2014 Error Rate (%)

The network architecture in this paper is quite different from VGGNet, ZFNet, and AlexNet. It contains **1×1 Convolution** at the middle of the network. And **global average pooling** is used at the end of the network instead of using fully connected layers. These two techniques are from another paper "Network In Network" (NIN) [6]. Another technique, called **inception module**, is to have different sizes/types of convolutions for the same input and stacking all the outputs.

And authors also mentioned that the idea of the name "**Inception**", is coming from NIN and a famous internet meme below: **WE NEED TO GO DEEPER**.

WE NEED TO GO DEEPER

ImageNet, is a dataset of over 15 millions labeled high-resolution images with around 22,000 categories. ILSVRC uses a subset of ImageNet of around 1000 images in each of 1000 categories. In all, there are roughly 1.2 million training images, 50,000 validation images and 100,000 testing images.

What we'll cover:

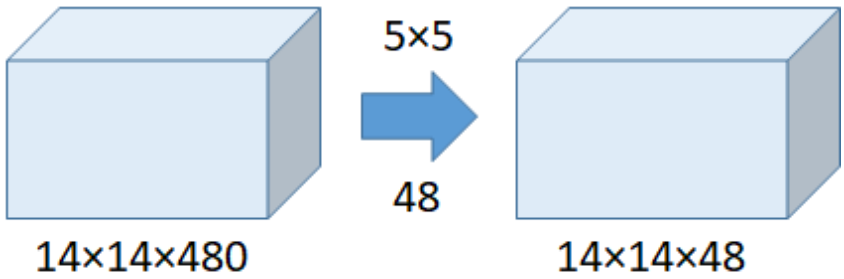
1. **The 1×1 Convolution**
2. **Inception Module**
3. **Global Average Pooling**
4. **Overall Architecture**
5. **Auxiliary Classifiers for Training**

6. **Testing Details**

1. The 1×1 Convolution

The 1×1 convolution is introduced by NIN [6]. 1×1 convolution is used with ReLU. Thus, originally, NIN uses it for introducing more non-linearity to increase the representational power of the network since authors in NIN believe data is in non-linearity form. **In GoogLeNet, 1×1 convolution is used as a dimension reduction module to reduce the computation. By reducing the computation bottleneck, depth and width can be increased.**

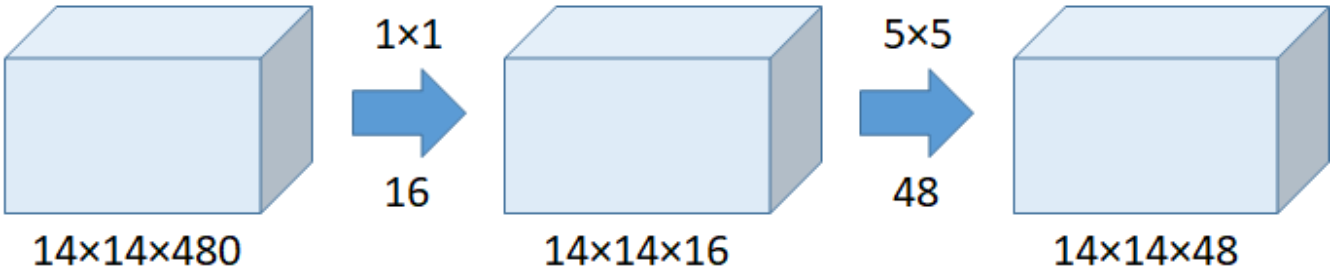
I pick a simple example to illustrate this. Suppose we need to perform 5×5 convolution **without the use of 1×1 convolution** as below:



Without the Use of 1×1 Convolution

Number of operations = (14×14×48)×(5×5×480) = 112.9M

With the use of 1×1 convolution:



With the Use of 1×1 Convolution

Number of operations for 1×1 = (14×14×16)×(1×1×480) = 1.5M
Number of operations for 5×5 = (14×14×48)×(5×5×16) = 3.8M

Total number of operations = 1.5M + 3.8M = 5.3M
which is much much smaller than 112.9M !!!!!!!!!!!!!!

Indeed, the above example is the calculation of **5×5 conv at inception (4a).**

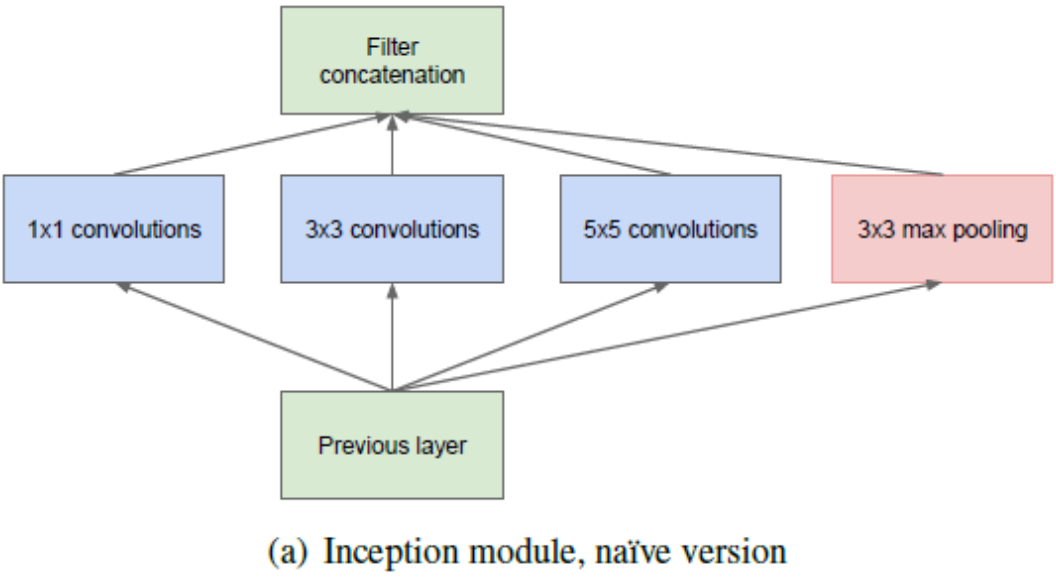
(We may think that, when dimension is reduced, actually we are working on the mapping from high dimension to low dimension in a non-linearity way. In contrast, for PCA, it performs linear dimension reduction.)

Thus, **inception module can be built without increasing the number of operationslargely compared the one without 1×1 convolution!**

> 1×1 convolution can help to reduce model size which can also somehow help to reduce the overfitting problem!!

2. Inception Module

The inception module (naive version, without 1×1 convolution) is as below:



(a) Inception module, naïve version

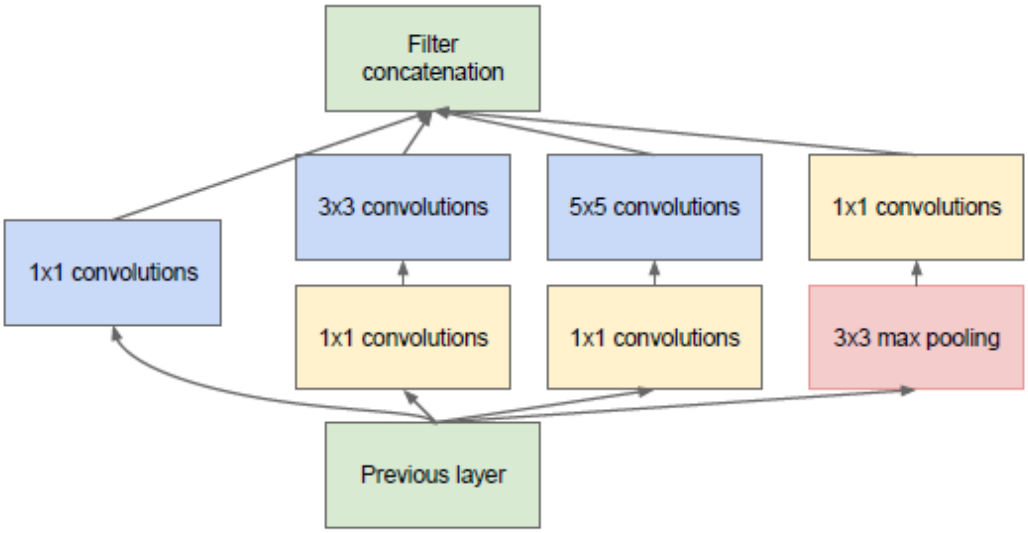
Inception Module (Without 1×1 Convolution)

Previously, such as AlexNet, and VGGNet, conv size is fixed for each layer.

Now, **1×1 conv**, **3×3 conv**, **5×5 conv**, and **3×3 max pooling** are done altogether for the previous input, and stack together again at output. **When image's coming in, different sizes of convolutions as well as max pooling are tried. Then different kinds of features are extracted.**

After that, all feature maps at different paths are concatenated together as the input of the next module.

However, without the 1×1 convolution as above, we can imagine how large the number of operation is!

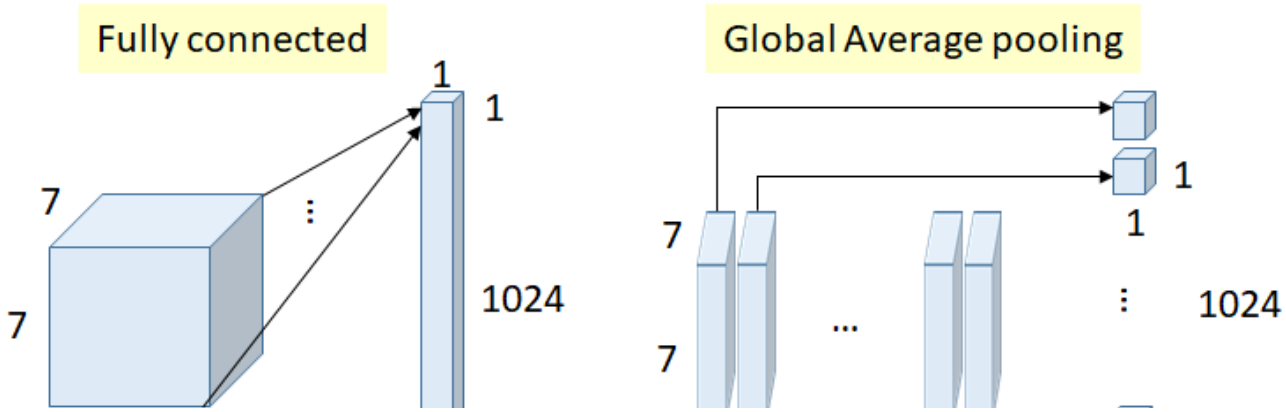


(b) Inception module with dimensionality reduction

Inception Module (With 1×1 Convolution)

Thus, 1×1 convolution is inserted into the inception module for dimension reduction!

3. Global Average Pooling



Fully Connected Layer VS Global Average Pooling

Previously, **fully connected (FC) layers** are used at the end of network, such as in AlexNet. All inputs are connected to each output.

Number of weights (connections) above = 7×7×1024×1024 = 51.3M

In GoogLeNet, global average pooling is used nearly at the end of network by averaging each feature map from 7×7 to 1×1, as in the figure above.

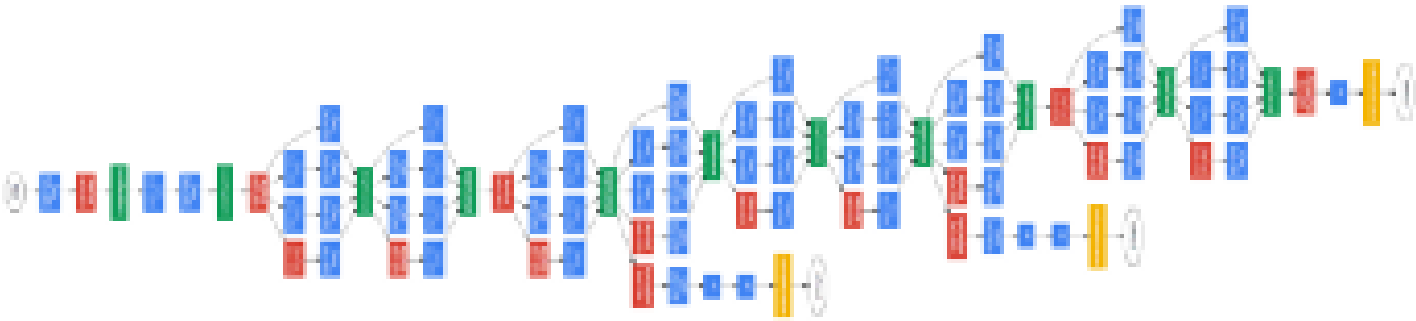
Number of weights = 0

And authors found that a move from FC layers to **average pooling improved the top-1 accuracy by about 0.6%.**

This is the idea from NIN [6] which can be **less prone to overfitting.**

4. Overall Architecture

After knowing the basic units as described above, we can talk about the overall network architecture.



GoogLeNet Network (From Left to Right)

There are 22 layers in total!

It is already a very deep model compared with previous AlexNet, ZFNet and VGGNet. (But not so deep compared with ResNet invented afterwards.) And we can see that **there are numerous inception modules connected together to go**

deeper. (There are some intermediate softmax branches at the middle, we will describe about them in the next section.)

Below is the details about the parameters if each layer. We actually can extend the example of 1×1 convolution to calculate the number of operations by ourselves. :)

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Details about Parameters of Each Layer in GoogLeNet Network (From Top to Bottom)

5. Auxiliary Classifiers for Training

As we can see there are some **intermediate softmax branches** at the middle, they are used for training only. These branches are auxiliary classifiers which consist of:

5×5 Average Pooling (Stride 3)
1×1 Conv (128 filters)
1024 FC
1000 FC
Softmax

The loss is added to the total loss, with weight 0.3.

Authors claim it can be used for combating gradient vanishing problem, also providing regularization.

And it is NOT used in testing or inference time.

6. Testing Details

7 GoogLeNet are used for ensemble prediction. This is already a kind of boosting approach from LeNet, AlexNet, ZFNet and VGGNet.

Multi-scale testing is used just like VGGNet, with shorter dimension of 256, 288, 320, 352. (4 scales)

Multi-crop testing is used, same idea but a bit different from and more complicated than AlexNet.

First, for each scale, it takes left, center and right, or top, middle and bottom squares (3 squares). Then, for each square, 4 corners and center as well as the resized square (6 crops) are cropped as well as their corresponding flips (2 versions) are generated.

The total is **4 scales×3 squares×6 crops×2 versions=144 crops/image**

Softmax probabilities are averaged over all crops.

Number of models	Number of Crops	Cost	Top-5 error	compared to base
1	1	1	10.07%	base
1	10	10	9.15%	-0.92%
1	144	144	7.89%	-2.18%
7	1	7	8.09%	-1.98%
7	10	70	7.62%	-2.45%
7	144	1008	6.67%	-3.45%

7 Models + 144 Crops

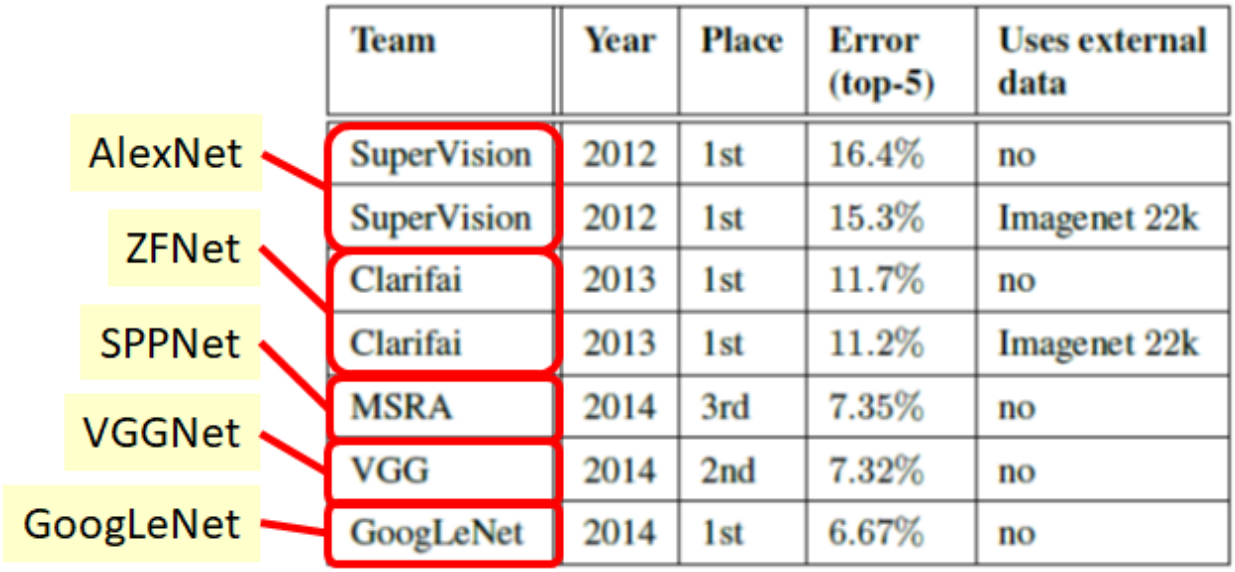
Ablation Study

With 7 models + 144 crops, the top-5 error is 6.67%.

Compared with 1 model + 1 crop, there are large reduction from 10.07%.

From this, we can observe that, **besides the network design**, the other stuffs like **ensemble methods, multi-scale and multi-crop approaches are also essential to reduce the error rate!!!**

And these techniques actually are not totally new in this paper!



Team	Year	Place	Error (top-5)	Uses external data
SuperVision	2012	1st	16.4%	no
SuperVision	2012	1st	15.3%	Imagenet 22k
Clarifai	2013	1st	11.7%	no
Clarifai	2013	1st	11.2%	Imagenet 22k
MSRA	2014	3rd	7.35%	no
VGG	2014	2nd	7.32%	no
GoogLeNet	2014	1st	6.67%	no

Comparison with State-of-the-art Approaches

Finally, GoogLeNet outperforms other previous deep learning networks, and won in ILSVRC 2014.

I will review other deep learning networks as well as inception versions later on. If interested, please also visit the reviews of LeNet [7], AlexNet [8] , ZFNet [9], and VGGNet [10].

References

1. [2015] [CVPR] [GoogLeNet]
   Going Deeper with Convolutions

2. [2014 ECCV] [ZFNet]
   Visualizing and Understanding Convolutional Networks

3. [2012 NIPS] [AlexNet]
   ImageNet Classification with Deep Convolutional Neural Networks

4. [2015 ICLR] [VGGNet]
   Very Deep Convolutional Networks for Large-Scale Image Recognition

5. [1998 Proc. IEEE] [LeNet-1, LeNet-4, LeNet-5, Boosted LeNet-4]
   Gradient-Based Learning Applied to Document Recognition

6. [2014 ICLR] [NIN]
   Network in Network

7. Review of LeNet-1, LeNet-4, LeNet-5, Boosted LeNet-4 (Image Classification)

8. Review of AlexNet, CaffeNet — Winner of ILSVRC 2012 (Image Classification)

9. Review of ZFNet — Winner of ILSVRC 2013 (Image Classification)

10. Review of VGGNet — 1st Runner-Up of ILSVLC 2014 (Image Classification)