An Efficient Deep Neural Network Architecture for ComputerVision

Monalisa Samal[1], Sudhir Kumar Sa[2] and D. K. Mohanty[3]

[1]Assistant Professor, Department of Electronics and Communication Engineering, Gandhi Institute For Technology (GIFT), Bhubaneswar
[2]Assistant Professor, Department of Electronics and Communication Engineering, Gandhi Engineering College, Bhubaneswar
[3]Assistant Professor, School of Computer Engineering, KIIT University, Bhubaneswar

Publishing Date: Oct 31, 2015

Abstract

Deep neural network is a rich family of methods, comprising of neural networks, probabilistic models, and different types of unsupervised and supervised feature learning algorithms. In the recent years deep learning methods have been applied in several fields, with computer vision being one of the most important application.In this paper, we propose Deep Cconvolutional Neural Nnetwork (DCNN) architecture. The main objective of this architecture is toimprove utilization of the computing resources inside the network. In this study, we increased the depth and width of the network while computational budget is kept unchanged. The architectural decisions were based on the Hebbian principle in order to optimize quality.

1. Introduction

Deep learning allows computational models of multiple processing layers to learn and represent data with multiple levels of abstraction mimicking how the brain perceives and understands multimodal information, thus implicitly capturing intricate structures of large-scale data.In the last three years, our object classification and de- tection capabilities have dramatically improved due to ad- vances in deep learning and convolutional networks [10]. One encouraging news is that most of this progress is not just the result of more powerful hardware, larger datasets andbiggermodels,butmainlyaconsequenceofnewideas, algorithms and improved network architectures. No new data sources were used, for example, by the top entries in the ILSVRC 2014 competition besides the classification datasetofthesamecompetitionfordetectionpurposes.Our GoogLeNet submission to ILSVRC 2014 actually uses 12 times fewer parameters than the winning architecture of Krizhevsky et al [9] from two years ago, while being significantly more accurate. On the object detection front, the biggest gains have not come from naive application ofbiggerandbiggerdeepnetworks,butfromthesynergyofdeep architecturesandclassicalcomputervision,liketheR-CNN algorithm by Girshick et al[6].

Another notable factor is that with the ongoing traction of mobile and embedded computing, the efficiency of our algorithms–especiallytheirpowerandmemoryuse–gains importance. It is noteworthy that the considerationsleading tothedesignofthedeeparchitecturepresentedinthispaper included this factor rather than having a sheer fixation on accuracynumbers.Formostoftheexperiments,themodels weredesignedtokeepacomputationalbudgetof1.5billion multiply-adds at inference time, so that the they do not end uptobeapurelyacademiccuriosity,butcouldbeputtoreal world use, even on large datasets, at a reasonablecost.

In this paper, we will focus on an efficient deep neural network architecture for computer vision, codenamed Inception, which derives its name from the Network in networkpaperbyLinetal[12]inconjunctionwiththefamous "we need to go deeper" internet meme [1]. In our case, the word "deep" is used in two different meanings: first of all, in the sense that we introduce a new level of organization in the form of the "Inception module" and also in the more directsenseofincreasednetworkdepth.Ingeneral,onecan view the Inception model as a logical culmination of [12] while taking inspiration and guidance from the theoretical work by Arora et al [2]. The benefits of the architectureare experimentally verified on the ILSVRC 2014classification anddetectionchallenges,whereitsignificantlyoutperforms the current state of theart.

2. Related Work

Starting with LeNet-5 [10], convolutional neural networks (CNN) have typically had a standard structure – stacked convolutional layers (optionally followed by con-

trast normalization and max-pooling) are followed by one ormorefully-connectedlayers.Variantsofthisbasicdesign are prevalent in the image classification literature and have yieldedthebestresultsto-dateonMNIST,CIFARandmost notably on the ImageNet classification challenge [9, 21]. For larger datasets such as Imagenet, the recent trend has been to increase the number of layers [12] and layer size[21,14],whileusingdropout[7]toaddresstheproblem ofoverfitting.

Despite concerns that max-pooling layers result in loss ofaccuratespatialinformation,thesameconvolutionalnetwork architecture as [9] has also been successfully employedforlocalization[9,14],objectdetection[6,14,18,5] and human pose estimation[19].

Inspired by a neuroscience model of the primate visual cortex, Serre et al. [15] used a series of fixed Gabor filters ofdifferentsizestohandlemultiplescales.Weuseasimilar strategy here. However, contrary to the fixed 2-layer deep model of [15], all filters in the Inception architecture are learned. Furthermore, Inception layers are repeated many times, leading to a 22-layer deep model in the case of the GoogLeNetmodel.

Network-in-Network is an approach proposed by Lin et al. [12] in order to increase the representational power of neural networks. In their model, additional 1×1 convolutional layers are added to the network, increasing its depth. We use this approach heavily in our architecture. However, in our setting, 1×1 convolutions have dual purpose: most critically,theyareusedmainlyasdimensionreductionmodules to remove computational bottlenecks, that would otherwise limit the size of our networks. This allows for not justincreasingthedepth,butalsothewidthofournetworks without a significant performancepenalty.

Finally, the current state of the art for object detectionis theRegionswithConvolutionalNeuralNetworks(R-CNN)methodbyGirshicketal.[6].R-CNNdecomposestheover- all detection problem into two subproblems: utilizing low- level cues such as color and texture in order to generateob- ject location proposals in a category-agnostic fashion and using CNN classifiers to identify object categories at those locations. Such a two stage approach leverages the accu- racy of bounding box segmentation with low-level cues, as wellasthehighlypowerfulclassificationpowerofstate-of-the-art CNNs. We adopted a similar pipeline in our detection submissions, but have explored enhancements in both stages, such as multi-box [5] prediction for higher object bounding box recall, and ensemble approaches for better categorization of bounding boxproposals.

3. Motivation and High LevelConsiderations

The most straightforward way of improving the performance of deep neural networks is by increasing their size. Thisincludesbothincreasingthedepth–thenumberofnet-

Figure 1: Two distinct classes from the 1000 classes of the ILSVRC 2014 classification challenge. Domain knowledge is required to distinguish between theseclasses.

work levels – as well as its width: the number of units at each level. This is an easy and safe way of training higher quality models, especially given the availability of a large amount of labeled training data. However, this simple solution comes with two majordrawbacks.

Bigger size typically means a larger number of parameters,whichmakestheenlargednetworkmorepronetooverfitting, especially if the number of labeled examples in the trainingsetislimited.Thisisamajorbottleneckasstrongly labeleddatasetsarelaboriousandexpensivetoobtain,often requiring expert human raters to distinguish between variousfine-grainedvisualcategoriessuchasthoseinImageNet (even in the 1000-class ILSVRC subset) as shown in Figure1.

The other drawback of uniformly increased network size is the dramatically increased use of computational resources. For example, in a deep vision network, if two convolutional layers are chained, any uniform increase in the number of their filters results in a quadratic increase of computation. If the added capacity is used inefficiently(for example, if most weights end up to be close to zero), then much of the computation is wasted. As the computational budget is always finite, an efficient distribution of computing resources is preferred to an indiscriminate increase of size,evenwhenthemainobjectiveistoincreasethequality ofperformance.

Afundamentalwayofsolvingbothoftheseissueswould betointroducesparsityandreplacethefullyconnectedlay- ers by the sparse ones, even inside the convolutions. Be- sides mimicking biological systems, this would also have theadvantageoffirmertheoreticalunderpinningsduetothe groundbreaking work of Arora et al. [2]. Their main resultstatesthatiftheprobabilitydistributionofthedatasetis representable by a large, very sparse deep neural network, then the optimal network topology can be constructedlayer after layer by analyzing the correlation statistics of the preceding layer activations and clustering neurons with highly correlated outputs. Although the strict mathematical proof requires very strong conditions, the fact that thisstatement

resonates with the well knownHebbian principle –neurons that fire together, wire together – suggests that the underlying idea is applicable even under less strict conditions, in practice.

Unfortunately, today's computing infrastructures are very inefficient when it comes to numerical calculation on non-uniform sparse data structures. Even if the number of arithmetic operations is reduced by 100×, the overhead of lookups and cache misses would dominate: switching to sparse matrices might not pay off. The gap is widened yet further by the use of steadily improving and highly tuned numerical libraries that allow for extremely fast dense matrix multiplication, exploiting the minute details of the underlying CPU or GPU hardware [16, 9]. Also,non-uniform sparse models require more sophisticated engineering and computing infrastructure. Most current vision oriented machinelearningsystemsutilizesparsityinthespatialdomain justbythevirtueofemployingconvolutions.However,convolutions are implemented as collections of dense connections to the patches in the earlier layer. ConvNets have traditionally used random and sparse connection tables in the feature dimensions since [11] in order to break the symmetry and improve learning, yet the trend changed back to full connections with [9] in order to further optimize parallel computation. Current state-of-the-art architectures for computer vision have uniform structure. The large number of filters and greater batch size allows for the efficient use of densecomputation.

This raises the question of whether there is any hope for a next, intermediate step: an architecture that makes use of filter-level sparsity, as suggested by the theory, but exploits our current hardware by utilizing computations on dense matrices. The vast literature on sparse matrix computations (e.g. [3]) suggests that clustering sparse matrices into relatively dense submatrices tends to give competitive performance for sparse matrix multiplication. It does not seemfar-fetchedtothinkthatsimilarmethodswouldbeutilized for the automated construction of non-uniform deeplearning architectures in the nearfuture.

The Inception architecture started out as a case studyfor assessingthehypotheticaloutputofasophisticatednetwork topology construction algorithm that tries to approximate a sparsestructureimpliedby[2]forvisionnetworksandcovering the hypothesized outcome by dense, readily available components. Despite being a highly speculative undertaking, modest gains were observed early on when compared with reference networks based on [12]. With a bit of tuning the gap widened and Inception proved to be especially useful in the context of localization and object detection as the base network for [6] and [5]. Interestingly, while most of the original architectural choices have been questioned and tested thoroughly in separation, they turned out to be close to optimal locally. One must be cautious though:although the Inception architecture has become a success for computer vision, it is still questionable whether this can be attributedtotheguidingprinciplesthathaveleadtoitsconstruction. Making sure of this would require a much more thorough analysis andverification.

4. ArchitecturalDetails

ThemainideaoftheInceptionarchitectureistoconsider howanoptimallocalsparsestructureofaconvolutionalvi- sion network can be approximated and covered by readily availabledensecomponents.Notethatassumingtranslation invariancemeansthatournetworkwillbebuiltfromconvolutional building blocks. All we need is to find the optimal local construction and to repeat it spatially. Arora et al. [2] suggestsalayer-bylayerconstructionwhereoneshouldanalyze the correlation statistics of the last layer and cluster them into groups of units with high correlation. Theseclusters form the units of the next layer and are connected to the units in the previous layer. We assume that each unit from an earlier layer corresponds to some region of the input image and these units are grouped into filter banks. In thelowerlayers(theonesclosetotheinput)correlatedunits would concentrate in local regions. Thus, we would end up with a lot of clusters concentrated in a single region and they can be covered by a layer of 1×1 convolutions in the next layer, as suggested in [12]. However, one can also expectthattherewillbeasmallernumberofmorespatially spreadoutclustersthatcanbecoveredbyconvolutionsover larger patches, and there will be a decreasing number of patches over larger and larger regions. In order to avoid patch-alignment issues, current incarnations of the Inception architecture are restricted to filter sizes 1×1, 3×3and 5×5; this decision was based more on convenience rather thannecessity.Italsomeansthatthesuggestedarchitecture is a combination of all those layers with their output filter banks concatenated into a single output vector forming the input of the next stage. Additionally, since pooling operations have been essential for the success of current convolutionalnetworks,itsuggeststhataddinganalternativeparallelpoolingpathineachsuchstageshouldhaveadditional beneficial effect, too (see Figure2(a)).

Asthese"Inceptionmodules"arestackedontopofeach other, their output correlation statistics are bound to vary: as features of higher abstraction are captured by higherlayers, their spatial concentration is expected to decrease.This suggests that the ratio of 3×3 and 5×5 convolutionsshould increase as we move to higherlayers.

One big problem with the above modules, at least inthis naïveform, is that even a modest number of 5×5 convolutions can be prohibitively expensive on top of a convolutionallayerwithalargenumberoffilters.Thisproblembecomes even more pronounced once pooling units areadded to the mix: the number of output filters equals to the number of filters in the previous stage. The merging of

output of the pooling layer with outputs of the convolutional lay- ers would lead to an inevitable increase in the number of outputs from stage to stage. While this architecture might cover the optimal sparse structure, it would do it very inef- ficiently, leading to a computational blow up within a few stages.

This leads to the second idea of the Inception architecture: judiciously reducing dimension wherever the computational requirements would increase too much otherwise. This is based on the success of embeddings: even low dimensionalembeddings might contain a lot of information about a relatively large image patch. However, embeddings represent information in a dense, compressed form and compressed information is harder to process. The representationshouldbekeptsparseatmostplaces(asrequired by the conditions of [2]) and compress the signals only whenevertheyhavetobeaggregatedenmasse.Thatis, 1×1 convolutions are used to compute reductions before the expensive 3×3 and 5×5 convolutions. Besides being usedasreductions,theyalsoincludetheuseofrectifiedlinearactivationmakingthemdual-purpose.Thefinalresultis depicted in Figure2(b).

In general, an Inception network is a network consisting of modules of the above type stacked upon each other, with occasional max-pooling layers with stride 2 to halve the resolution of the grid. For technical reasons (memory efficiency during training), it seemed beneficial to start using Inception modules only at higher layers while keeping thelowerlayersintraditionalconvolutionalfashion.Thisis notstrictlynecessary,simplyreflectingsomeinfrastructural inefficiencies in our currentimplementation.

A useful aspect of this architecture is that it allows for increasing the number of units at each stage significantly without an uncontrolled blow-up in computational com- plexity at later stages. This is achieved by the ubiquitous useofdimensionalityreductionpriortoexpensiveconvolutions with larger patch sizes. Furthermore, the design fol- lows the practical intuition that visual information should be processed at various scales and then aggregated so that thenextstagecanabstractfeaturesfromthedifferentscales simultaneously.

The improved use of computational resources allowsfor increasing both the width of each stage as well as the numberofstageswithoutgettingintocomputationaldifficulties. One can utilize the Inception architecture to create slightly inferior, but computationally cheaper versions of it. We have found that all the available knobs and levers allowfor a controlled balancing of computational resourcesresulting in networks that are 3 10 faster than similarly perform- ing networks with non-Inception architecture, however this requires careful manual design at thispoint.

5. GoogLeNet

By the"GoogLeNet" name we refer to the particular incarnation of the Inception architecture used in our submission for the ILSVRC 2014 competition. We also used one deeper and wider Inception network with slightly superior quality,butaddingittotheensembleseemedtoimprovethe resultsonlymarginally.Weomitthedetailsofthatnetwork, as empirical evidence suggests that the influence of the exact architectural parameters is relatively minor. Table 1 illustratesthemostcommoninstanceofInceptionusedinthe competition. This network (trained with different image-patchsamplingmethods)wasusedfor6outofthe7models in ourensemble.

All the convolutions, including those inside the Inception modules, use rectified linear activation. The size ofthe receptivefieldinournetworkis224224inthe RGBcolor space with zero mean. "#3 3 reduce" and "#5 5 reduce" stands for the number of 1 1 filters in the reduction layer used before the 3 3 and 5 5 convolutions. One can see the number of 1 1 filters in the projection layer after the built-in max-pooling in the pool proj column. All these reduction/projection layers use rectified linear activation as well.

Thenetworkwasdesignedwithcomputationalefficiency and practicality in mind, so that inference can be run onindividual devices including even those with limited computational resources, especially with low-memoryfootprint.

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								

Table 1: GoogLeNet incarnation of the Inception architecture.

The network is 22 layers deep when counting only layers withparameters(or27layersifwealsocountpooling).The overallnumberoflayers(independentbuildingblocks)used for the construction of the network is about 100. The exact number depends on how layers are counted by the machine learning infrastructure. The use of average pooling before theclassifierisbasedon[12],althoughourimplementation hasanadditionallinearlayer.Thelinearlayerenablesusto easily adapt our networks to other label sets, however it is usedmostlyforconvenienceandwedonotexpectittohave a major effect. We found that a move from fully connected layers to average pooling improved the top-1 accuracy by about 0.6%, however the use of dropout remainedessential even after removing the fully connectedlayers.

Given relatively large depth of the network, the ability to propagate gradients back through all the layers in an effective manner was a concern. The strong performance of shallower networks on this task suggests that the features produced by the layers in the middle of the network should be very discriminative. By adding auxiliary classifiers connected to these intermediate layers, discrimination in the lower stages in the classifier was expected. This was thought to combat the vanishing gradient problem while providing regularization. These classifiers take the form of smaller convolutional networks put on top of the out- put of the Inception (4a) and (4d) modules. During train- ing, their loss gets added to the total loss of the network with a discount weight (the losses of the auxiliary classi- fiers were weighted by 0.3). At inference time, these auxil- iarynetworksarediscarded.Latercontrolexperimentshave shown that the effect of the auxiliary networks is relatively minor (around 0.5%) and that it required only one of them to achieve the sameeffect.

The exact structure of the extra network on the side, in- cluding the auxiliary classifier, is as follows:

- An average pooling layer with 5×5 filter size and stride 3, resulting in an 4×4×512 output for the(4a), and 4×4×528for the (4d)stage.
- A 1 1 convolution with 128 filters for dimension re- ductionand rectified linearactivation.
- A fully connected layer with 1024 units and rectified linearactivation.
- A dropout layer with 70% ratio of droppedoutputs.

• A linear layer with softmax loss as the classifier (predictingthesame1000classesasthemainclassifier,but removed at inferencetime).

A schematic view of the resulting network is depictedin Figure3.

6. TrainingMethodology

GoogLeNet networks were trained using the DistBe- lief [4] distributed machine learning system using mod- est amount of model and data-parallelism. Although we used a CPU based implementation only, a rough estimate suggests that the GoogLeNet network could be trained to convergence using few high-end GPUs within a week, the main limitation being the memory usage. Our training used asynchronous stochastic gradient descent with 0.9 momentum[17],fixedlearningrateschedule(decreasingthelearn- ing rate by 4% every 8 epochs). Polyak averaging [13] was used to create the final model used at inferencetime.

Image sampling methods have changed substantially over the months leading to the competition, and already convergedmodelsweretrainedonwithotheroptions,sometimes in conjunction with changed hyperparameters, such as dropout and the learning rate. Therefore, it is hard to give a definitive guidance to the most effective single way totrainthesenetworks.Tocomplicatemattersfurther,some ofthemodelsweremainlytrainedonsmallerrelativecrops, others on larger ones, inspired by [8]. Still, one prescrip- tion that was verified to work very well after the competi- tion, includes sampling of various sized patches of the im- age whose size is distributed evenly between 8% and100% of the image area with aspect ratio constrained to the inter- val[$\frac{3}{4}$, $\frac{4}{3}$]. Also, we found that the photometricdistortions of Andrew Howard [8] were useful to combat overfitting to the imaging conditions of training data.

7. ILSVRC 2014 Classification Challenge Setup andResults

The ILSVRC 2014 classification challenge involves the taskofclassifyingtheimageintooneof1000leaf-nodecategories in the Imagenet hierarchy. There are about 1.2 million images for training, 50,000 for validation and 100,000 images for testing. Each image is associated with one ground truth category, and performance is measured based on the highest scoring classifier predictions. Two numbers are usually reported: the top-1 accuracy rate, which compares the ground truth against the first predicted class, and the top-5 error rate, which compares the ground truth against the first 5 predicted classes: an image is deemed correctly classified if the ground truth is among the top-5, regardless of its rank in them. The challenge uses the top-5error rate for rankingpurposes.

We participated in the challenge with no external data used for training. In addition to the training techniques aforementionedinthispaper,weadoptedasetoftechniques duringtestingtoobtainahigherperformance,whichwedescribenext.

1. We independently trained 7 versions of the same GoogLeNet model (including one wider version), and performed ensemble prediction with them. These models were trained with the same initialization (even with the same initial weights, due to an oversight) and learning rate policies. They differed only in sampling methodologies and the randomized input imageorder.

2. Duringtesting,weadoptedamoreaggressivecropping approach than that of Krizhevsky et al. [9]. Specif- ically, we resized the image to 4 scales where the shorter dimension (height or width) is 256, 288, 320 and 352 respectively, take the left, center and right square of these resized images (in the case of portrait images, we take the top, center and bottom squares). For each square, we then take the 4 corners and the center 224 224 crop as well as the square resizedto 224 224, and their mirrored versions. This leads to 4 3 6 2 = 144 crops per image. A similar ap- proach was used by Andrew Howard [8] in the pre- vious year's entry, which we empirically verified to performslightlyworsethantheproposedscheme.We notethatsuchaggressivecroppingmaynotbenecessaryinrealapplications,asthebenefitofmorecrops becomesmarginalafterareasonablenumberofcrops arepresent(aswewillshowlateron).

3. The softmax probabilities are averaged over multiple crops and over all the individual classifiers to obtain the final prediction. In our experiments we analyzed alternative approaches on the validation data, such as maxpoolingovercropsandaveragingoverclassifiers, but they lead to inferior performance than the simple averaging.

In the remainder of this paper, we analyze the multiple factorsthatcontributetotheoverallperformanceofthefinal submission.

Our final submission to the challenge obtains a top-5 errorof6.67%onboththevalidationandtestingdata,ranking the first among other participants. This is a 56.5% relative reduction compared to the SuperVision approach in 2012, and about 40% relative reduction compared to the previous year's best approach (Clarifai), both of which usedexternal data for training the classifiers. Table 2 shows the statistics of some of the top-performing approaches over the past 3 years.

We also analyze and report the performance of multiple testing choices, by varying the number of models and the

Team	Year	Place	Error (top-5)	Uses external data
SuperVision	2012	1st	16.4%	no
SuperVision	2012	1st	15.3%	Imagenet22k
Clarifai	2013	1st	11.7%	no
Clarifai	2013	1st	11.2%	Imagenet22k
MSRA	2014	3rd	7.35%	no
VGG	2014	2nd	7.32%	no
GoogLeNet	2014	1st	6.67%	no

Table 2: Classification performance.

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%

Table 3: GoogLeNet classification performance break down.

number of crops used when predicting an image in Table 3. When we use one model, we chose the one with the lowest top-1 error rate on the validation data. All numbers are re-portedonthevalidationdatasetinordertonotoverfitto the testing datastatistics.

8. ILSVRC 2014 Detection Challenge Setup andResults

The ILSVRC detection task is to produce bounding boxesaroundobjectsinimagesamong200possibleclasses. Detected objects count as correct if they match the class of the groundtruth and their bounding boxes overlap by at least 50% (using the Jaccard index). Extraneous detections count as false positives and are penalized. Contrary to the classificationtask,eachimagemaycontainmanyobjectsor none, and their scale may vary. Results are reported using the mean average precision (mAP). The approach taken by GoogLeNetfordetectionissimilartotheR-CNNby[6],but isaugmentedwiththeInceptionmodelastheregionclassi-fier. Additionally, the region proposal step is improved by combining the selective search [20] approach with multi- box [5] predictions for higher object bounding box recall. In order to reduce the number of false positives, thesuper-

Team	Year	Place	mAP	external data	ensemble	approach
UvA-Euvision	2013	1st	22.6%	none	?	Fisher vectors
Deep Insight	2014	3rd	40.5%	ImageNet1k	3	CNN
CUHK DeepID-Net	2014	2nd	40.7%	ImageNet1k	?	CNN
GoogLeNet	2014	1st	43.9%	ImageNet1k	6	CNN

Table 4: Comparison of detection performances. Unreported values are noted with question marks.

pixel size was increased by 2×. This halves the proposals comingfromtheselectivesearchalgorithm.Weaddedback 200 region proposals coming from multi-box [5] resulting, in total, in about 60% of the proposals used by [6], while increasing the coverage from 92% to 93%. The overall effect of cutting the number of proposals with increased coverage is a 1% improvement of the mean average precision for the single model case. Finally, we use an ensemble of6 GoogLeNets when classifying each region. This leads to an increase in accuracy from 40% to 43.9%. Note that contrary to R-CNN, we did not use bounding box regression due to lack oftime.

We first report the top detection results and show the progress since the first edition of the detection task. Compared to the 2013 result, the accuracy has almost doubled. The top performing teams all use convolutional networks. We report the official scores in Table 4 and common strategies for each team: the use of external data, ensemblemodels or contextual models. The external data is typically the ILSVRC12 classification data for pre-training a model that islaterrefinedonthedetectiondata.Someteamsalsomen- tion the use of the localization data. Since a good portion of the localization task bounding boxes are not included in the detection dataset, one can pre-train a general bounding box regressor with this data the same way classification is used for pre-training. The GoogLeNet entry did not use the localization data forpretraining.

InTable5,wecompareresultsusingasinglemodelonly. The top performing model is by Deep Insight and surprisinglyonly improves by 0.3 points with an ensemble of 3 models while the GoogLeNet obtains significantlystronger results with theensemble.

9. Conclusions

Ourresultsyieldasolidevidencethatapproximatingthe expectedoptimalsparsestructurebyreadilyavailabledense buildingblocksisaviablemethodforimprovingneuralnetworks for computer vision. The main advantage of this method is a significant quality gain at a modest increase of computational requirements compared to shallower and narrowerarchitectures.

Our object detection work was competitive despite not utilizing context nor performing bounding box regression, suggesting yet further evidence of the strengths of the Inception architecture.

Team	mAP	Contextual model	Bounding box regression
Trimps-Soushen	31.6%	no	?
Berkeley Vision	34.5%	no	yes
UvA-Euvision	35.4%	?	?
CUHK DeepID-Net2	37.7%	no	?
GoogLeNet	38.02%	no	no
Deep Insight	40.2%	yes	yes

Table 5: Single model performance for detection.

For both classification and detection, it is expected that similar quality of result can be achieved by much more expensive non-Inception-type networks of similar depth and width. Still, our approach yields solid evidence that moving to sparser architectures is feasible and useful idea in general. This suggest future work towards creating sparser and more refined structures in automated ways on thebasis of [2], as well as on applying the insights of the Inception architecture to otherdomains.

References

[1] Know your meme: We need to godeeper. http://knowyourmeme.com/memes/we-need-to-go-deeper. Accessed: 2014-09-15.

[2] S. Arora, A. Bhaskara, R. Ge, and T. Ma. Provable boundsforlearningsomedeeprepresentations. *CoRR*, abs/1310.6343,2013.

[3] U.V.Çatalyürek,C.Aykanat,andB.Uçar.On two-dimensional sparse matrix partitioning: Mod- els, methods, and a recipe. *SIAM J. Sci. Comput.*, 32(2):656–683, Feb.2010.

[4] J.Dean,G.Corrado,R.Monga,K.Chen,M.Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, Q. V. Le, and A. Y. Ng. Large scale distributed deep networks. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *NIPS*, pages 1232–1240. 2012.

[5] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *CVPR*,2014.

[6] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and PatternRecognition,2014.CVPR2014.IEEEConference on*,2014.

[7] G. E. Hinton, N. Srivastava, A.Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580,2012.

[8] A. G. Howard. Some improvements on deep convolutional neural network based image classification. *CoRR*, abs/1312.5402,2013.

[9] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks.In*AdvancesinNeuralInformationProcessing Systems 25*, pages 1106–1114,2012.

[10] Y.LeCun,B.Boser,J.S.Denker,D.Henderson,R.E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551, Dec.1989.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[12] M. Lin, Q. Chen, and S. Yan. Network innetwork. *CoRR*, abs/1312.4400, 2013.

[13] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM J. ControlOptim.*, 30(4):838–855, July1992.

[14] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229,2013.

[15] T.Serre,L.Wolf,S.M.Bileschi,M.Riesenhuber,and T. Poggio. Robust object recognition with cortex-like mechanisms.*IEEE Trans. Pattern Anal. Mach.Intell.*, 29(3):411–426,2007.

[16] F. Song and J. Dongarra. Scaling up matrix computations on shared-memory manycore systems with1000 cpu cores. In *Proceedings of the 28th ACM InternationalConferenceonSupercomputing*,ICS'14,pages 333–342, New York, NY, USA, 2014.ACM.

[17] I.Sutskever,J.Martens,G.E.Dahl,andG.E.Hinton. On the importance of initialization and momentum in deeplearning.In*ICML*,volume28of*JMLRProceedings*, pages 1139–1147. JMLR.org,2013.

[18] C.Szegedy,A.Toshev,andD.Erhan.Deepneuralnetworks for object detection. In C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, editors, *NIPS*, pages 2553–2561,2013.

[19] A. Toshevand C. Szegedy. Deeppose: Human pose estimation via deep neural networks. *CoRR*, abs/1312.4659,2013.

[20] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders. Segmentation as selective search for object recognition. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 1879–1886, Washington, DC, USA, 2011. IEEE ComputerSociety.

[21] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In D. J. Fleet, T.Pajdla, B. Schiele, and T. Tuytelaars, editors, *ECCV*, volume 8689 of *Lecture Notes in Computer Science*, pages 818–833. Springer,2014.