IMAGENET Large Scale Visual Recognition Challenge 2013 (ILSVRC2013)

Introduction History Data Tasks Timetable Citation[new] Organizers Sponsors Contact

Results of ILSVRC2013

Detection Classification Classification+Localization Team information Per-class results[new]

Task 1: Detection

Legend:
Dark grey background = outside training data;
Light grey background = not participating in competition (as requested by authors)
Valid submissions = entries which participated in competition and did not use outside training data

Team name	Comment	mean AP	**Number of categories won (only valid submissions)**	Number of categories won (all submissions)
UvA-Euvision	Run 2 (with prior)	0.22581	130	56
NEC-MU	Regionlets with HOG, LBP, Covariance feature and Neural Patterns, Nerual Patterns extractor uses a publicly avaliable image classification model	0.208954	--	35
UvA-Euvision	Run 3	0.208183	--	12
NEC-MU	Regionlets with HOG, LBP and Covariance feature, trained with boosting on train dataset, results merged by precision alignment using validation data	0.19617	25	13
OverFeat - NYU	1 ConvNet (trained with CLS data for pre-training, then supplied DET data only)	0.194009	--	55
UvA-Euvision	Run 1	0.192062	--	3
NEC-MU	Regionlets with HOG, LBP and Covariance feature, trained with boosting on train+val dataset,	0.191785	35	24
Toronto A	Detector conv net also predicts the aspect ratio of the bounding box in addition to the class label. The output is branched at the penultimate layer of the conv net.	0.114595	6	1
Toronto A	Multiplicative gating with 5 parts-based conv net detector. The learning algorithm is the same as before.	0.105838	0	0
SYSU_Vision	DPM without context	0.104501	3	1
GPU_UCLA	partial submission, no outside training data	0.098338	0	0
Toronto A	Incorporates multiplicative gating with a separate streaming classifier conv network. The separate stream provides context and helps to reduce overfitting by reducing the scores.	0.0839	1	0
	This models have 1 root conv net and 4 additional conv			

Toronto A	net detecting 4 parts, with scanning window approach and NMS, does not use any outside training data.	0.080738	0	0
Toronto A	This model uses 1 root conv net with linear svm on top. Features are learned from the positive images. Learning is performed using SGD with minibatches of 128.	0.077761	0	0
SYSU_Vision	DPM with CNN context	0.07545	0	0
Delta	200 classes + 1 background, no outside training data	0.060772	0	0
Delta	200 classes, no outside training data	0.060541	0	0
Delta	200 classes + 200 backgrounds, no outside training data	0.051104	0	0
UIUC-IFP	Convnet for object detection.	0.010489	--	0

Task 2: Classification

Legend:
Dark grey background = outside training data

Team name	Comment	**Error**
Clarifai	Multiple models trained on the original data plus an additional model trained on 5000 categories.	0.11197
Clarifai	Multiple models trained on the original data plus an additional model trained on other 1000 category data.	0.11537
Clarifai	Average of multiple models on original training data.	0.11743
Clarifai	Another attempt at multiple models on original training data.	0.1215
Clarifai	Single model trained on original data.	0.12535
NUS	adaptive non-parametric rectification of all outputs from CNNs and refined PASCAL VOC12 winning solution, with further retraining on the validation set.	0.12953
NUS	adaptive non-parametric rectification of all outputs from CNNs and refined PASCAL VOC12 winning solution.	0.13303
ZF	5 models (4 different architectrues) trained on original data.	0.13511
Andrew Howard	This is an ensemble of convolutional neural networks combining multiple transformations for training and testing and models operating at different resolutions.	0.13555
Andrew Howard	This method explores re weighting the predictions from different data transformation and ensemble members in the previous submission.	0.13564
ZF	5 models trained on original data, 1 big.	0.13748
ZF	5 models trained on original data, 1 long.	0.13894
ZF	5 same models +1 different models trained on original data.	0.13934
NUS	weighted sum of all outputs from CNNs and refined PASCAL VOC12 winning solution.	0.13985
ZF	5 same sized models trained on original data.	0.14079
OverFeat - NYU	7 ConvNet voting (trained using only supplied CLS data)	0.14182
UvA-Euvision	Main run	0.14291
NUS	weighted sum of outputs from one large CNN and five CNNs with 6-convolutional layers.	0.1502
Adobe	Averaged over 6 convolutional neural network. We use image saliency to obtain 9 crops from original images and combine them with the standard 5 multiview crops. No outside training data are used.	0.15193
VGG	a combination of a single deep Fisher network and a single deep convolutional neural network; no outside training data	0.15245
OverFeat -		

NYU	1 ConvNet (trained using only supplied CLS data)	0.15675
Adobe	Averaged over 6 convolutional neural network. We use the standard 5 multiview crops. No outside training data are used.	0.15963
CognitiveVision	Convolution Deep Neural Network with dropout, improved multi view test, and hierarchical classify	0.16052
CognitiveVision	Convolution Deep Neural Network with dropout, improved multi view test	0.16086
UvA-Euvision	Showcase mobile run	0.16586
VGG	a single deep convolutional neural network (similar to [Krizhevsky, 2012], but with less convolutional filters and with additional jittering); no outside training data	0.177
decaf	Decaf reference implementation, using one single CNN network.	0.19231
IBM Multimedia Team	estimation on prediction set	0.207
IBM Multimedia Team	baseline with CNN only	0.20788
Deep Punx	Several averaged Deep Convolutional Neural Networks	0.20926
Deep Punx	Deep Convolutional Neural Network with Dropout	0.21588
Minerva-MSRA	very large Convolutional Neural Network	0.21666
Minerva-MSRA	Convolutional Neural Network with LWTA/Maxout non-linearity	0.21705
Minerva-MSRA	Convolutional Neural Network with adaptive learning rate and dropout fraction	0.22178
NUS	traditional framework based on PASCAL VOC12 winning solution with extension of high-order parametric coding.	0.22389
Minerva-MSRA	Convolutional Neural Network with adaptive network topology	0.22783
VGG	a single Deep Fisher Network (accepted at NIPS 2013); no outside training data	0.23075
Deep Punx	Deep Convolutional Neural Network with DropConnect	0.23743
MIL	weighted sum 1	0.24426
MIL	weighted sum 3	0.24726
Orange	baseline results, learned feature , stochastic gradient, NO outside training data	0.25168
BUPT-Orange	Run2: Softmax, crossmap	0.25188
Orange	learning rate to 0.00001, NO outside training data	0.25194
BUPT-Orange	Run1: Softmax, samemap	0.25232
MIL	weighted sum 2	0.25323
MIL	weighted sum 4	0.25357
Orange	drop out blur1, NO outside training data	0.25467
Orange	drop out blur 2, NO outside training data	0.26183
Trimps-Soushen1	Using only supplied training data, single model.	0.26204
Trimps-Soushen1	Combine three models.	0.26204
Trimps-Soushen1	Using only supplied training data, with probability max pooling, single model.	0.26264
Deep Punx	Deep Convolutional Neural Network with additional data augmentation - rotations and scaling	0.26395
MIL	weighted sum 5	0.26642
Minerva-MSRA	Convolutional Neural Network with skip-level connection	0.26661

Orange	drop out blur 3, NO outside training data	0.26667
BUPT-Orange	Run3: Multi-SVM, crossmap	0.27306
IBM Multimedia Team	baseline with low level feature	0.66302
QuantumLeap	15 features (see abstract). RVM. No outside training data.	0.82015
Deep Punx	Deep Convolutional Neural Network with DLSVM instead of softmax layer	0.99521

Task 3: Classification+Localization

Team name	Comment	**Error**
OverFeat - NYU	1 ConvNet (trained using only supplied CLS+LOC data)	0.298772
VGG	Weakly-supervised localisation based on saliency maps; trained from image labels only; no outside training data	0.464242

Team information (more details at ILSVRC2013 workshop)

Team name	Team members	Abstract
Adobe	Hailin Jin, Adobe Zhe Lin, Adobe Jianchao Yang, Adobe Tom Paine, UIUC	Our algorithm is based on the University of Toronto NIPS 2012 paper. We use deep convolutional neural networks trained on RGB images using Dropout. We modify the network architecture to have more filters and connections. We train 6 networks with different settings. At test time, we use image saliency to obtain 9 crops from original images and combine them with the standard 5 multiview crops. We do not use any training data outside the challenge data.
Andrew Howard	Andrew Howard - Andrew Howard Consulting	We investigate techniques to improve on the state of the art convolutional neural network models from last year's competition. There are two main areas that we focused on. The first area was to explore additional transformations of the data for training and testing. When training models, we add additional data translations that extend into the pixels that are cropped out in the training pipeline from last years winning submission. We also add more color manipulations. This improves on the base model and allows us to build larger models without over fitting. When testing, we use the additional translations and also scalings to generate more diverse views of the data to improve predictions. The second area focused on improving the ensemble prediction when using multiple neural network models by including models trained at different resolutions. We took trained neural networks and fined tuned them to make predictions at higher resolutions in order to add complementary predictions and improve the overall ensemble prediction. By fine tuning already trained models we cut training time down on the higher resolution models by more than one half. We have also investigated weighting predictions from different transformations and models. The current (unweighted) submission has a top 5 error rate of 0.1417 and top 1 error rate of 0.3470 (0.1407 and 0.3457 for the weighted version) on the validation set using only competition data. We intend on adding more multiple resolution models to the ensemble for the extended deadline.
	Chong Huang, Beijing University of Posts and Telecommunications Yunlong Bian, Beijing University of Posts and	Task 2: Classification. Our team have submitted 3 runs: Run1_Softmax,

BUPT-Orange	Telecommunications Hongliang Bai, Orange Labs International Center Beijing Bo Liu, Beijing University of Posts and Telecommunications Yanchao Feng, Beijing University of Posts and Telecommunications Yuan Dong, Beijing University of Posts and Telecommunications	Run2_Softmax, Run3_MultiSVM. Our architectures have similar baseline, including 5 convolutional layer, 3 overlapping max/average pooling layer, 3 fully-connected layer. There are slight difference among these runs. In Run1_Softmax, the first and second convolutional layer is followed by the Local response normalization layer (same map).In Run2_Softmax, this kind of normalization layer is replaced by the Local response normalization layer (across map). We choose the softmax as the cost function in Run1_Softmax and Run2_Softmax. In the Run3_MultiSVM, the multi-SVM layer is choosen as the last layer instead of softmax layer. We train our model in one GPU for one week. The test images are augmented by translation transformation. The sum of all of transformed images is ranked as the final results.
Clarifai	Matthew Zeiler, Clarifai	A large deep convolutional network is trained on the original data to classify each of the 1,000 classes. The only preprocessing done to the data is subtracting a per-pixel mean. To augment the amount of training data, the image is downsampled to 256 pixels and a random 224 pixel crop is taken out of the image and randomly flipped horizontally to provide more views of each example. Additionally, the dropout technique of Hinton et al. "Improving neural networks by preventing co-adaptation of feature detectors" was utilized to further prevent overfitting. The architecture contains 65M parameters trained for 10 days on a single Nvidia GPU. By using a novel visualization technique based on the deconvolutional networks of Zeiler et. al, "Adaptive Deconvolutional Networks for Mid and High Level Feature Learning", it became clearer what makes the model perform, and from this a powerful architecture was chosen. Multiple such models were averaged together to further boost performance.
CognitiveVision	Kuiyuan Yang, Microsoft Research Yalong Bai, Harbin Institute of Technology Yong Rui, Microsoft Research	With the increase number of categories, image classification task is moved from the basic level to subordinate level (e.g., there are 120 breeds of dogs in ILSVRC 2013). Though biologically-inspired Deep Neural Network (DNN) has achieved great success in image classification task, it still cannot well distinguish categories at subordinate level. We use a cognitive psychology inspired image classification scheme using Deep Neural Network (DNN). Analogy to the learning process of human being, DNN firstly learns to classify the basic-level categories then learns to classify categories at the subordinate level for fine-grained object recognition.
decaf	Yangqing Jia, Jeff Donahue, Trevor Darrell UC Berkeley	Decaf is our open-source reference implementation for deep learning. Our submission reproduces the model presented by Alex Krizhevsky et al. in NIPS 2012 that won the ILSVRC 2012 challenge. We followed the same network architecture and training protocol adopted by Krizhevsky et al. with the only difference that we did not expand the training data by manipulating pixel colors, which accounts for the 1% accuracy difference in the original paper and our submission. The purpose of our submission is to provide and promote a publicly available, easy to use implementation for state-of-the-art deep learning approaches. Preliminary results from our group have already shown promising results on transfer learning and object detection, and we hope decaf could further help expand the availability of deep learning algorithms in computer vision.

		We have made the decaf code publicly available, and will release our pre-trained model for the submission soon. More details could be found at the following URLS: http://decaf.bekeleyvision.org/ (online demo) http://arxiv.org/abs/1310.1531 (decaf techinical report) http://arxiv.org/abs/1311.2524 (on object detection)
Deep Punx	Evgeny Smirnov, Denis Timoshenko, Alexey Korolev Saint Petersburg State University	Or base model is a deep convolutional neural network, similar to [1]. It was trained on NVIDIA GPU. We used Rectified Linear Units, five convolutional, three fully-connected and three max-pooling layers. To reduce overfitting we used Dropout and some data augmentation. In our second neural network we used another regularization method - Dropconnect [2]. In the third network we used different data augmentation method - random rotations and scaling. In the fourth network we replaced last layer (softmax) with DLSVM [3]. Finally, we averaged predictions of several neural networks. [1] ImageNet Classification with Deep Convolutional Neural Networks, Krizhevsky, A., Sutskever, I. and Hinton, G. E., NIPS 2012 [2] Regularization of Neural Network using DropConnect, Li Wan, Matthew Zeiler, Sixin Zhang, Yann LeCun, Rob Fergus, ICML 2013 [3] Deep Learning using Linear Support Vector Machines, Yichuan Tang, ICML 2013 Workshop in Challenges in Representation Learning
Delta	Che-Rung Lee, NTHU Hwann-Tzong Chen, NTHU Hao-Ping Kang, NTHU Tzu-Wei Huang, NTHU Ci-Hong Deng, NTHU Hao-Che Kao, NTHU	We use a generic object detector to find candidate locations of objects. The generic object detector combines the outputs of different approaches including salient object segmentation, "what is an object", and Otsu's algorithm. According to the validation results, we find that the generic object detector can achieve 0.76 mAP on detecting the objects of the 200 categories of interest, without knowing the specify categories of the detected objects. Based on the candidate object locations found by the generic object detector, we use a pre-trained deep net on GPUs to classify the candidate object locations and decide the categories of the objects.
GPU_UCLA	Yukun Zhu, Jun Zhu, Alan Yuille	Our method utilizes a compositional model for object detection. This model consists of one root node and several leaf nodes. The nodes in first layer capture viewpoint variations of each object, while the second layer captures HOG features for the entire object. The third and following layers decompose the object into several parts, and they capture fine-grained features for each object. To reduce the model complexity, we use CUDA programming techniques to accelerate our model. The inference for each image takes in average 1 seconds for each object class.
	Zhicheng Yan, University of Illinois at Urbana-Champaign Liangliang Cao, IBM Watson Research Center John R Smith, IBM Watson Research Center Noel Codella, IBM	With an unprecedentedly large scale, image classification task in ILSVRC2013 is known as the most challenging one in vision community. To achieve high performance, we leverage two models: support vector machines and convolutional neural network (CNN) hierarchical feature. On one hand, we train a linear SVM classifier using low level features from IMARS systems. On the other hand, starting from raw image pixel data, we build up 3 deep convolutional neural networks, all of which

IBM Multimedia Team	Watson Research Center Michele Merler, IBM Watson Research Center Sharath Pankanti, IBM Watson Research Center Sharon Alpert, IBM Haifa Research Center Yochay Tzur, IBM Haifa Research Center	consist of 5 convolutional layers and 3 fully-connected layers. We use several techniques (e.g. randomized data augmentation, dropout, weight decaying) to facilitate training. Our training is based on an efficient GPU implementation, which allows us to complete training in 10 days on a single GPU. As the final step, we average the predicted label probabilities from SVM and CNN to further improve our prediction. For training data, we only use the training and validation set from ILSVRC2013.
MIL	Masatoshi Hidaka Chie Kamada Yusuke Mukuta Naoyuki Gunji Yoshitaka Ushiku Tatsuya Harada from The Univ. of Tokyo	Local descriptors were transformed into fisher-based feature vectors. Linear classifiers were trained by averaged passive-aggressive algorithm. The test images were annotated by weighted sum of scores from the linear classifiers.
Minerva-MSRA	Tianjun Xiao, Peking University and Microsoft Research Minjie Wang, SJTU and Microsoft Research Jianpeng Li, XiDian University and Microsoft Research Yalong Bai, Harbin Institute of Technology and Microsoft Research Jiaxing Zhang, Microsoft Research Kuiyuan Yang, Microsoft Research Chuntao Hong, Microsoft Research Zheng Zhang, Microsoft Research	We approach the classification task by leveraging a new training platform that we built, called Minerva[1]. Minerva expresses a training procedure as a series of matrix operations, in a Matlab-like imperative and procedural programming style, resulting in compact code. The system automatically converts the code into an efficient internal representation for execution during runtime. Without changing a line of code, a training procedure runs on top of modern laptop/workstation, high-end server and server cluster, with and without GPU acceleration. On single GPU, Minerva runs approximately 30~40 % faster than cuda-Convnet. The programmability power of Minerva allows us to rapidly experiment new alternatives. Our baseline is last year's winning entry from University of Toronto [4], a deep convolutional net without pre-train. We replace the ReLU unit with more powerful piecewise linear units, including LWTA (local-winner-take-all[2]) and Maxout[3]. We also add skip-level weights to compensate the loss of low-level details resulted from aggressive max-pooling in the baseline network. We experiment with bigger model that the Minerva implementation enables, upwards to 1.44 billion connections and ~120 million parameters. Finally, we adjust learning rate and dropout rate adaptively to accelerate convergence and to cope with overfitting. [1]Wang, Minjie and Xiao, Tianjun and Li, Jianpeng and Zhang, Jiaxing and Hong, Chuntao and Wu, Ming and Shao, Bin and Zhang, Zheng, Minerva: a scalable and highly efficient training platform for deep learning. Under submission to Eurosys 2014. [2]Srivastava, Rupesh Kumar and Masci, Jonathan and Kazerounian, Sohrob and Gomez, Faustino and Schmidhuber, J{\"u}rgen, Compete to Compute, Technical Report No. IDSIA-04-13, Dalle Molle Institute for Artificial Intelligence, 2013. [3]Goodfellow, Ian J and Warde-Farley, David and Mirza, Mehdi and Courville, Aaron and Bengio, Yoshua, Maxout networks, arXiv preprint arXiv:1302.4389, 2013. [4]Krizhevsky, Alex and Sutskever, Ilya and Hinton, Geoff, Imagenet

		classification with deep convolutional neural networks, NIPS 2012.
NEC-MU	Xiaoyu Wang, NEC Labs America Miao Sun, University of Missouri Tianbao Yang, NEC Labs America Yuanqing Lin, NEC Labs America Tony X. Han, University of Missouri Shenghuo Zhu, NEC Labs America	The detection approach is built upon the recently published detection framework: Regionlets generic object detector[1]. It firstly generates object hypotheses using image segmentation. The Regionlets detector is trained on these object hypotheses. We investigated HOG, LBP, Covariance and Neural Patterns for the Regionlets framework. These features are combined in the low level and a boosting process is employed to pick up efficient representations. The final detection result is re-ranked using context. We extended the deformable part based model with LBP and PCA to complement the detection of very small objects for submission(1-3). [1] X. Wang, M. Yang, S. Zhu, Y. Lin, “Regionlets for Generic Object Detection”, ICCV 2013
NUS	LIN Min*, CHEN Qiang*, DONG Jian, HUANG Junshi, XIA Wei, YAN Shuicheng. (* indicates equal contribution.) National University of Singapore.	Task2 without extra data Adaptive-NPR: In this submission, we propose a so-called “adaptive non-parametric rectification” method to instance-specifically and non-parametrically correct/rectify the outputs from multiple shallow and deep experts/classifiers for obtaining more accurate prediction. Basically for each sample in the training and validation sets, we have a pair of outputs-from-experts and ground-truth label. For a testing sample, we use non-parametric method (regularized kernel regression or k-NN based on outputs-from-experts) to determine the affinities between the test sample and its auto-selected training/validation samples, and then the affinities are utilized to fuse the ground-truth labels (as well as the outputs-from-experts of both testing sample and selected samples) of these selected samples to produce a rectified prediction. More importantly, the optimal values of some tunable parameters (e.g. kernel parameter in kernel regression, tradeoff between ground-truth labels and outputs-from-experts in fusing stage, etc.) vary significantly for different samples. In this submission, we first determine the optimal values for these tunable parameters of each training/validation sample, and then for a test sample, we determine its values for these tunable parameters by referring to its k-NN neighbors, and the instance-adaptive values thus enhance the capability to provide more robust rectification. In this submission, we use two types of experts, namely, the conventional shallow SVM-based methods (based on our PASCAL VOC2012 winning solutions, with the new extension of high-order parametric coding in which the first and second order parameters of the adapted GMM for each instance are both considered) and the recently well-developed deep CNN methods. For CNN methods, we consider 6 convolutional layers and also a very large neural network with doubled nodes in convolutional layers of [Alex et al. 2012] (Thank Alex for sharing the core code). All these shallow and deep methods/experts serve as the foundation of the proposed adaptive non-parametric rectification framework.
	Hongliang BAI, Orange Labs International Center Beijing Lezi Wang, Beijing University of Posts and	

Orange	Telecommunications Shusheng Cen, Beijing University of Posts and Telecommunications YiNan Liu, Beijing University of Posts and Telecommunications Kun Tao, Orange Labs International Center Beijing Wei Liu, Orange Labs International Center Beijing Peng Li, Orange Labs International Center Beijing Yuan Dong, Orange Labs International Center Beijing	Deep learning has achieved big success in Imagenet LSVRC2012 by Hinton's team. In this year, we also designed one deep convolutional neural network and run them on the NVidia K5000 GPU workstation. The basic structure is convolution layers concatenated with full connected layers. Experiments have been conducted in the different layer number, structure, dropout algorithm, classifier type (SVM or Softmax), optimal algorithm(gradient descent, stochastic gradient, LBFGS). In the LSVRC2012 evaluation dataset, the top-5 error rate can less than 0.3.
OverFeat - NYU	Pierre Sermanet, David Eigen, Michael Mathieu, Xiang Zhang, Rob Fergus, Yann LeCun	Our submission is based on an integrated framework for using Convolutional Networks for classification, localization and detection. We use a multiscale and sliding window approach, efficiently implemented within a ConvNet. This not only improves classification performance, but naturally allows the prediction of one or more objects' bounding boxes within the image. The same basic framework was applied to all three tasks. For the classification task, we vote among different views presented to the network. For localization and detection, each sliding window classification is refined using a regressor trained to predict bounding boxes; we produce final predictions by combining the regressor outputs.
		We harness the power of RVM (relevance vector machine) and multiple feature sources to train a large-scale multiclass classifier. It attains very high sparsity, fast to train, and performs comparably with, if not better than, the current state-of-the-art. With as few as 54 relevance vectors in total (less than 0.0043% of all the ~1.3M training data), it already achieves a training set accuracy and top 5-hit rate of 7.692% and 20.28%, respectively. We find that using multiple feature sources makes the RVM even sparser while performing comparably with current state-of-the-art. In a small subset of 6 categories randomly chosen from ILSVRC 2013 (with 1300 x 6 = 7800 training images and 50 x 6 = 300 validation images), we find: 1. RVM with multiple feature sources achieves a validation accuracy of 70% with < 20 relevance vectors, and achieves 83% with < 120 relevance vectors. 2. RVM with a single feature source achieves a validation accuracy of 76% with approximately 300 relevance vectors. 3. SVM (optimally tuned by cross validation on training set for its C parameter) with single feature source achieves a validation accuracy of 79% with > 1000 support vectors. Unlike is the case with SVM, the total running time of our training algorithm is approximately linear in the number of relevance vectors eventually found. This fact, together with extreme sparsity, directly translates to very fast training time of our classifier. The training of all the

QuantumLeap	Henry Shu (Self-employed) Jerry Shu (Student in Troy High School, Fullerton, CA)	1.2M images are done in a desktop computer equipped with one Intel i7-4770K CPU, one GeForce GTX 780 GPU, and 32GB DDR3 RAM. On all the ~1.3M images, the empirical training time is < 300 seconds per relevance vector. This includes all hard disk I/O. In the literature, RVM has several desirable properties. Firstly, the number of relevance vectors it uses is very small. In binary classification, it is typically an order of magnitude smaller than the number of support vectors used in an SVM. Secondly, it is inherently a multiclass classifier and gives a probability output for each class in a test data point. Thirdly, the classifier has no parameters to tune. In this team, we extend RVM in a multiclass setting and carry out the classification in the (infinite dimensional) kernel space rather than the feature space. We use 15 features sources. They are 1. tiny_image (See CloudCV) 2. all_color (See CloudCV) 3. spatial pyramid of visual words (Lazebnik et al 2006) 4. denseSIFT (See CloudCV) 5. bag of visual words 6. gist (Olivia and Torralba 2001) 7. hog2x2 (See CloudCV) 8. line_hists (See CloudCV) 9. texton (Similar to 5., See CloudCV) 10. ssim (See CloudCV) 11. sparse_sift (See CloudCV) 12. lbphf (See CloudCV) 13. geo_map8x8 (See CloudCV) 14. geo_color (See CloudCV) 15. geo_texton (See CloudCV)
SYSU_Vision	Xiaolong Wang, Sun Yat-Sen University, China.	For Task 1 (Detection): The basic method is based on Pedro Felzenszwalb's deformable part models. I implemented the release 4.0 version code by C++, including the training and testing parts. I then extended it to a MPI version. I trained the models in a distributed system with 200 computers. A convolutional neural network is also trained to do classification. The network was trained on the detection dataset with 200 categories. I used the classification results to provide context information and rescored the detection results from DPM. Due to the time limitation, I have not done the bounding box prediction.
Toronto A	Yichuan Tang*, Nitish Srivastava*, Ruslan Salakhutdinov. (* = equal contribution)	The base model of our approach is the convolutional net classifiers for detection by using a scanning window approach. The conv net is first trained to classify all 200 class with the object aligned and centered. Low level features are shared for all classes, achieving computational efficiency. 200 One-vs-rest SVM classifiers sits atop the conv net. For detection, we employ the standard scanning window approach at multiple scales. Non-maximal suppression is used to arrive at the final detection bounding boxes. In addition to this base approach, we have also explored part-based approach where instead of 1 "root" conv net, we trained 4 additional conv net to recognize the top/left/bot/right parts of objects in the positive set. This is same as DPMs but without deformations. We have also trained a detection net as a separate gating network to improve performance by

	University of Toronto.	reducing false positive detections. Finally, we have another architecture where the aspect ratio of the bounding box is predicted using the conv net, improving IOU scores.
Trimps-Soushen1	Jie Shao, Xiaoteng Zhang, Yanfeng Shang, Wenfei Wang, Lin Mei, Chuanping Hu. (The Third Research Institute of the Ministry of Public Security, P.R. China)	For efficiency considerations, we trained a fast deep convolutional neural network model based on cuda-convnet. The model was trained on single NVIDIA GPU for two days. It has 35 million parameters, consists of five convolutional layers, and three full-connected layers. To improve the performance, several methods were used, include fast dropout, probability max pooling, novel model combination.
UIUC-IFP	Thomas Paine (UIUC) Kevin Shih (UIUC) Thomas Huang (UIUC)	[This method uses outside training data] We are interested in how a neural network approach with minimal engineering compares to existing detection methods. As such, our framework is very similar to the one recently proposed by [Girshick et. al., 2013]. However, our method is rather stripped down in comparison. We first pre-train a convolutional neural network [Krizhevsky et. al., 2012] on the full Imagenet Challenge 2012 Classification dataset, achieving 41% error on 1000 classes. We then remove the final layer and replace it with random weights to fine-tune a classifier with 200 classes on the positive detection images. This achieves 38% error. During training, we take the smallest square crop that encloses as much of the ground-truth bounding box as possible while still fitting within the image. This method allows the neural network to use context when scoring a detection, but makes sacrifices in terms of bounding box localization. Our method also makes no use of negative training data. Object detection is scored using only the final layer of a 200 class neural network. This can only hurt the final score. We are curious how well the neural network would fare on its own. Lastly, our method at the moment uses no region proposal system. Instead, at test time, we extract 128 square crops at four scales (50px, 100px, 200px, 300px) and resize them to fit the input of the neural network.
		For task 1, the ILSVRC2013 detection task on 200 classes, we submit two runs. Our runs utilize a new way of efficient encoding. The method is currently under submission, therefore we do not include identifying details on this part. The submission utilizes selective search (Uijlings et al. IJCV 2013) to create on many candidate boxes per image. These boxes are

UvA-Euvision	Koen E. A. van de Sande Daniel H. F. Fontijne Cees G. M. Snoek Harro M. G. Stokman Arnold W. M. Smeulders University of Amsterdam Euvision Technologies	represented by extracting densely sampled color SIFT descriptors (van de Sande et al, PAMI 2010) at multiple scales. The box is then encoded with our new efficient coding. The method is faster than bag-of-words with hard assignment and outperforms it in terms of accuracy. Each box is encoded with a multi-level spatial pyramid. Training follows a standard negative mining procedure based on the previous work. The first run is context-free. The 200 models are trained independently of one another. The second run utilizes a convolutional network, trained on the DET dataset, to compute a prior for the presence of an object in the image. No (pre-)training on other datasets has been performed. For task 2, the ILSVRC2013 classification task on 1,000 classes, we submit two runs. Our showcase run performs all evaluations of the test set on an iPhone 5s at a rate of 2 images per second, whereas on the iPhone 4 it has a performance of 1 image per 10 seconds. The results in the main run are based on the fusion of convolutional networks. The networks are compatible to the networks that won this task last year (Krizhevsky et al, NIPS 2012), where our networks have 76M free parameters. The parameters have been trained for 300 epochs on a single GPU. For training in both runs we have used the ImageNet 1,000 dataset. No (pre-)training on other datasets has been performed. Demo At the ILSVRC2013 workshop we will release an app in the App Store performing instant interactive photo classification (take a picture, see the top 5 ImageNet scores). This app uses the same engine as our Impala app that is already available at: https://itunes.apple.com/us/app/impala/id736620048 . The Impala app user interface was designed for the experience that the iPhone works for you, but can still be optimized. The current results reflect the match of the training data with the personal data on the iPhone.
VGG	Karen Simonyan Andrea Vedaldi Andrew Zisserman Visual Geometry Group, University of Oxford	In the classification challenge, we used a combination of two deep architectures: the deep Fisher vector network and the deep convolutional network. The deep Fisher network can be seen as the extension of the conventional Fisher vector representation to deep architectures. It incorporates several layers of Fisher vector encoders (we used two), placed on top each other. To prevent the explosion in the number of parameters, we injected discriminatively trained dimensionality reduction between the layers. As the low-level features, we used off-the-shelf dense SIFT and colour features. The classification was performed using calibrated one-vs-rest linear SVMs. The paper, describing the Fisher network architecture, has been accepted to publication at NIPS 2013. As the second deep architecture, we employed the state-of-the-art convolutional neural network, similar to the one used by Krizhevsky et al. for their ILSVRC-2012 submission. In our case, we used less convolutional filters, but employed an additional type of the training set augmentation, which models the occlusion effects. We present the results of both classification methods, computed independently, as well as their combination, obtained by the multiplication of the class posteriors. We did not employ any outside data for training.

		In the classification with localisation challenge, we used the preliminary version of a novel weakly-supervised method, based on the saliency estimation using the deep convolutional network (CNN). We did not utilise the provided training bounding boxes, but used exactly the same CNN, which was used in the classification challenge, to obtain the class-specific image saliency maps. Object bounding boxes were then estimated from these maps.
ZF	Matthew D Zeiler, New York University Rob Fergus, New York University	The approach is based on a combination of large convolutional networks with a range of different architectures. The choice of architectures was assisted by visualization of model features' using a deconvolutional network, as described in Zeiler et. al "Adaptive Deconvolutional Networks for Mid and High Level Feature Learning", ICCV 2011. Each model is trained on a single Nvidia GPU for more than one week. Data is augmented by resizing the images to 256x256 pixels and then selecting random 224x224 pixel crops and horizontal flips from each example. This data augmentation is combined with the Dropout method of Hinton et al. ("Improving neural networks by preventing co-adaptation of feature detectors"), which prevents overfitting in these large networks.