Tracking with Deep Neural Networks

Hao Guan(管皓)

School of Computer Science
Fudan University
2014-12-29

Question

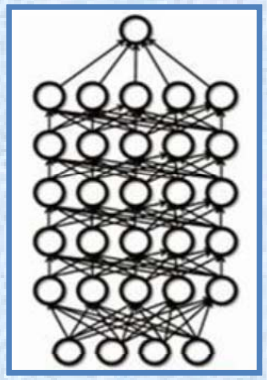
Humans do not need any information or prior knowledge of the object before tracking.

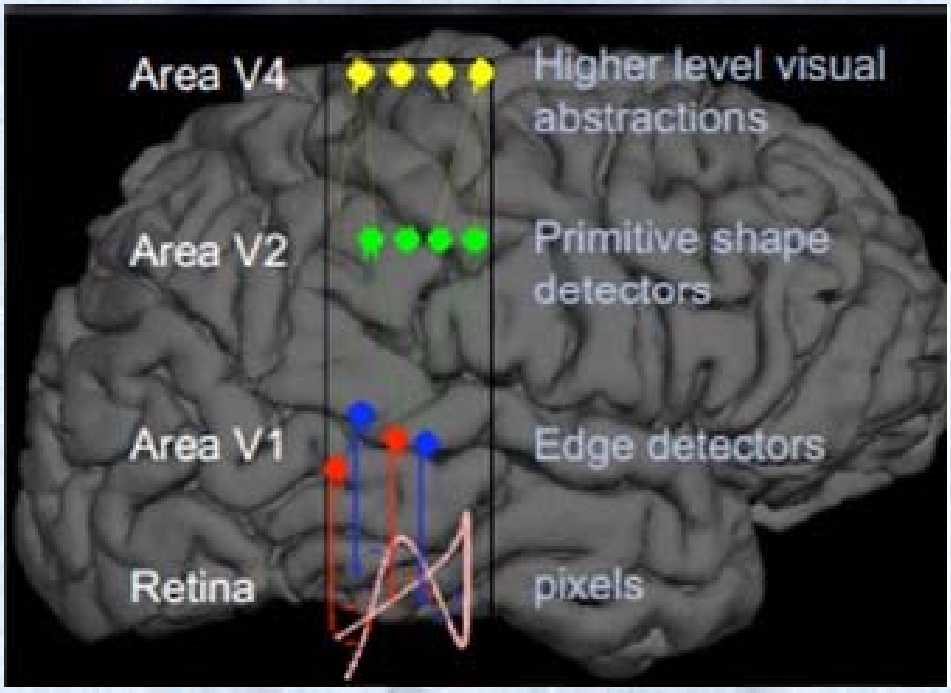
Question

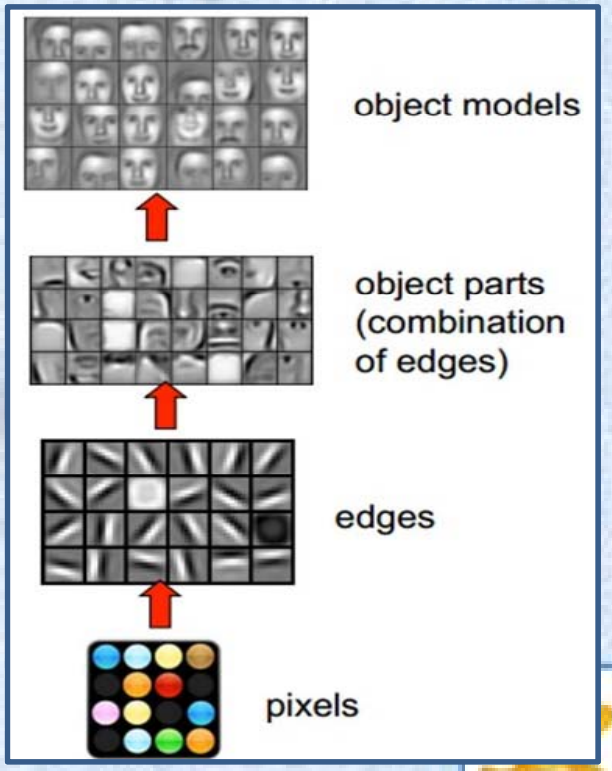
Humans are able to track unknown object with only a few images.

Question

Deep learning methods have become the dominant artificial vision system for object detection and classification.
It is a fusion of bio-inspired and neuromorphic vision models.







Great Success in classification and detection

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

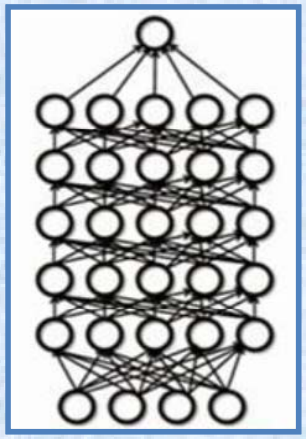
Abstract

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

Question

Deep learning methods have become the dominant artificial vision system for object detection and classification.

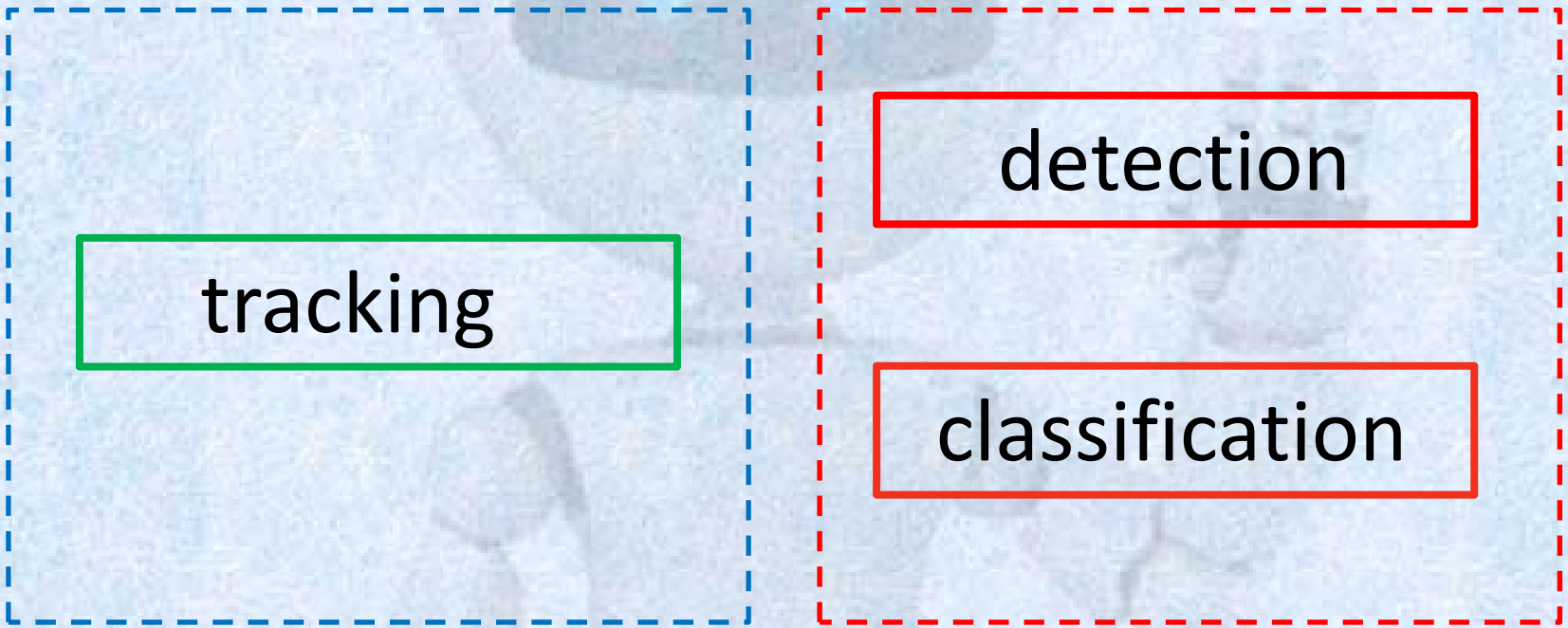
Then can the deep learning methods be applied to tracking?



Visual tracking

Difference and difficulty

- The components of a vision system



Difference and difficulty

- Real time applications

  Constrained by the network size, dataset volume, optimization algorithm.

  The network should support at least ten frames-per-second operation on commercial hardware.

detection

tracking

classification

Difference and difficulty

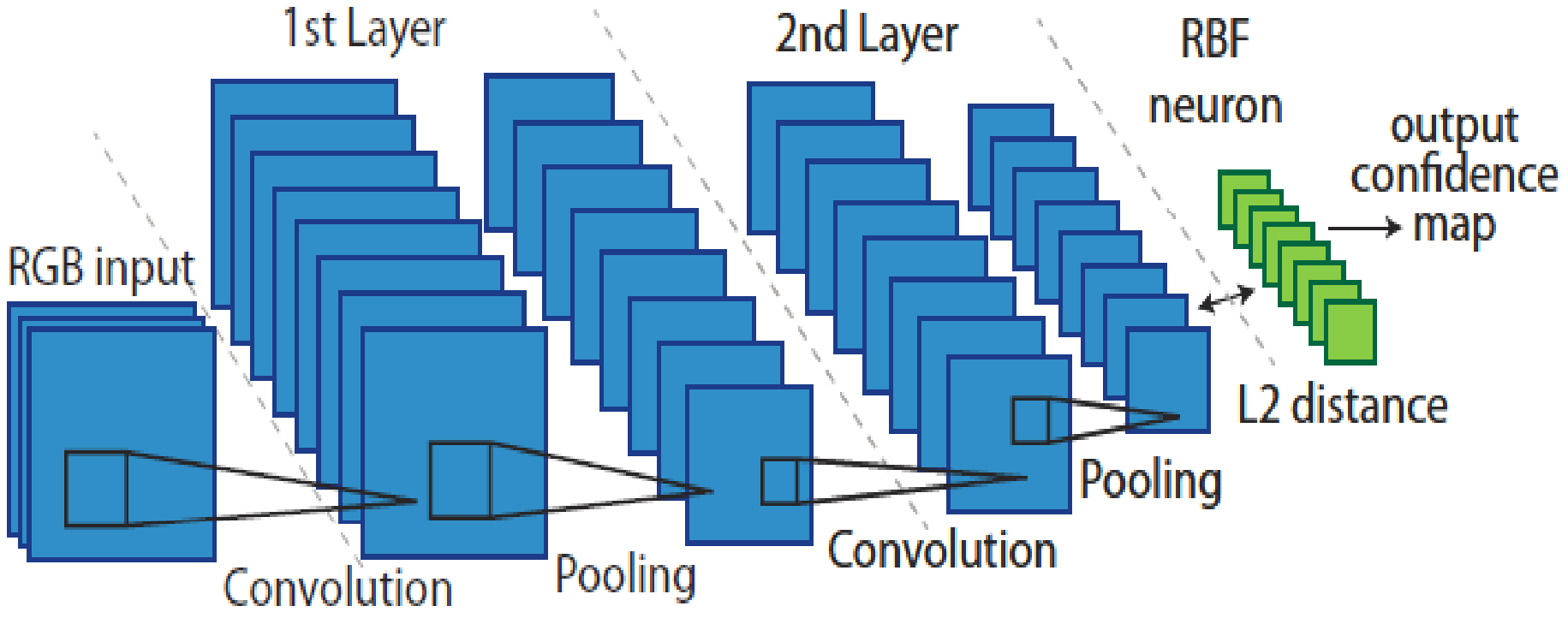
- General purpose VS. specific tasks — e.g. robotic vision
  It is sometimes not interesting to train the network to perform only on one dataset, when the levels would not carry over another dataset or real-world images.

detection

tracking

classification

Structure of the deep tracking system



Algorithm

Input Video frames $(I^{(1)}, ..., I^{(t)})$

Rectangle $r^{(1)}$ of object in the first frame

Output Rectangle $(r^{(2)}, ..., r^{(t)})$ of object

Initialization (for frame $I^{(1)}$):

1) Normalize frame $I^{(1)}$ by subtracting the mean and dividing the standard deviation.
2) Extract a small patch $X^{(1)}$ from rectangle $r^{(1)}$.
3) Feed the patch $X^{(1)}$ to the network and compute the single output vector $Y^{(1)} = f_{ConvNet}\left(X^{(1)}\right)$ where $Y^{(1)} \in \Re^{k \times 1}$.
4) Generate the first neuron, centered at $Y^{(1)}$, in RBFN to save the feature vector $Y^{(1)}$ for positive prototype.

Algorithm

Tracking (for each frame $I^{(t)}, t > 1$)

1) Normalize frame $I^{(t)}$ by subtracting the mean and dividing the standard deviation.
2) Slice the whole frame $I^{(t)}$ into small patches $X_{ij}^{(t)}$ (i, j are indexes for row and column).
3) Feed the small patches $X_{ij}^{(t)}$ to the network and compute the output vectors $Y_{ij}^{(t)} = f_{ConvNet}\left(X_{ij}^{(t)}\right)$ where $Y_{ij}^{(t)} \in \Re^{k \times 1}$.
4) Produce confidence map based on the distance between inputs $Y_{ij}^{(t)}$ and the neuron $Y^{(1)}$ in RBFN.
5) Draw a rectangle $r^{(t)}$ where the peak of the map is larger than threshold τ.

Convolutional neural network

- Spatial Convolution

$$h_{ijk}^{(t)} = tanh\left(W_k * X_{ij}^{(t)} + b_k\right)$$

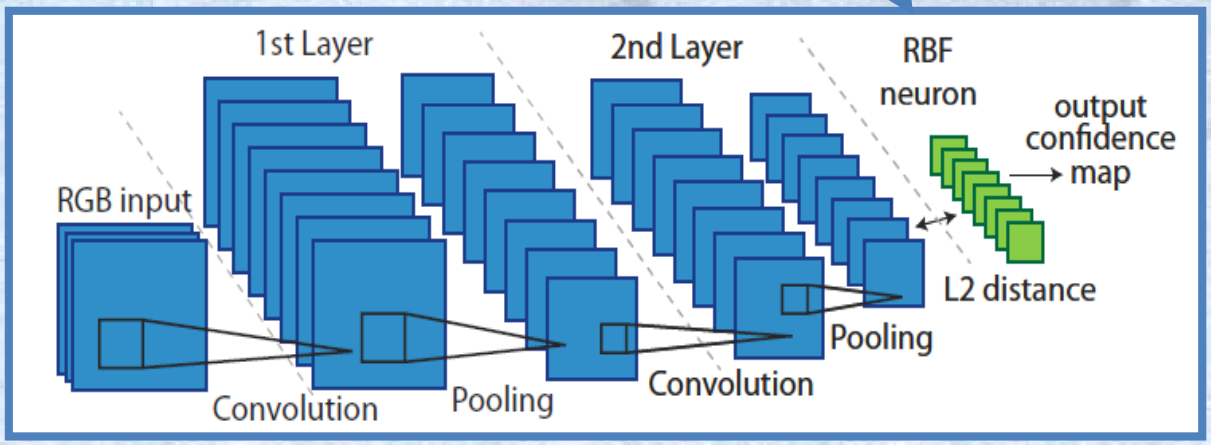
- Pooling

$$y_k^{(t)} = \sqrt{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left\|h_{ijk}^{(t)}\right\|_2^2}$$

Radial Basis Function Network

RBFN computes the distance from the reference vectors defined as centroids of neurons and finds the closest match.

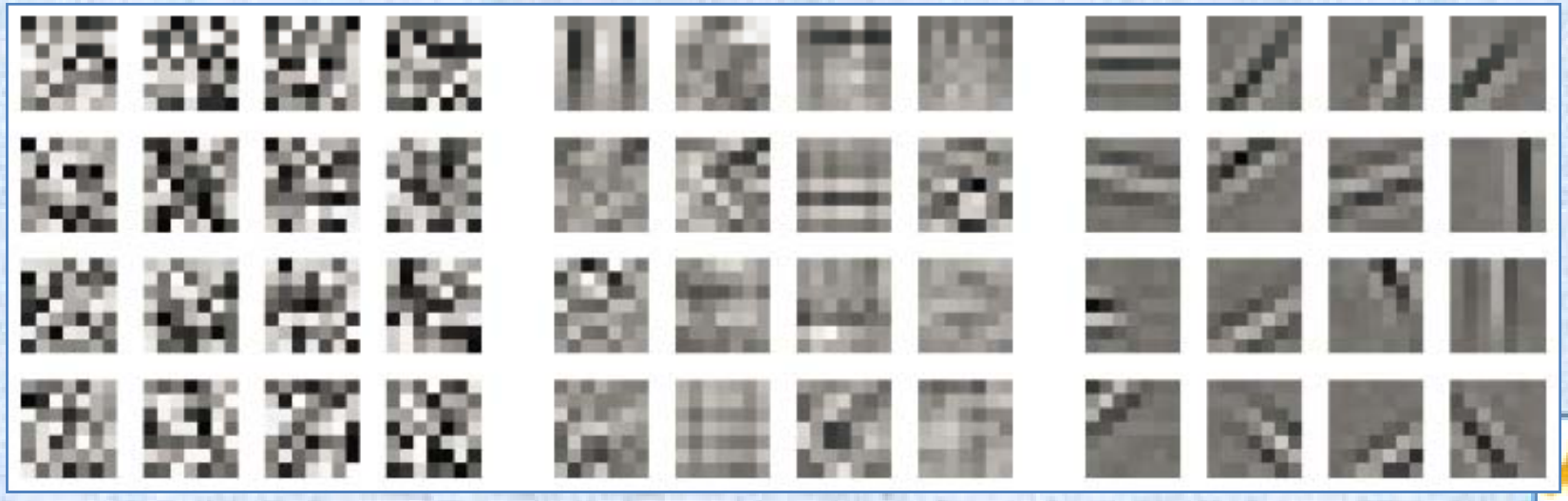
$$z(Y) = \sum_{i=1}^{N} w_i \phi(\|Y - Y_i\|)$$



Network Properties

Three different kernels for experiment , all are 7 × 7:

- Parameters with random initialization
- Parameters with supervised learning method
- Parameters with unsupervised learning method

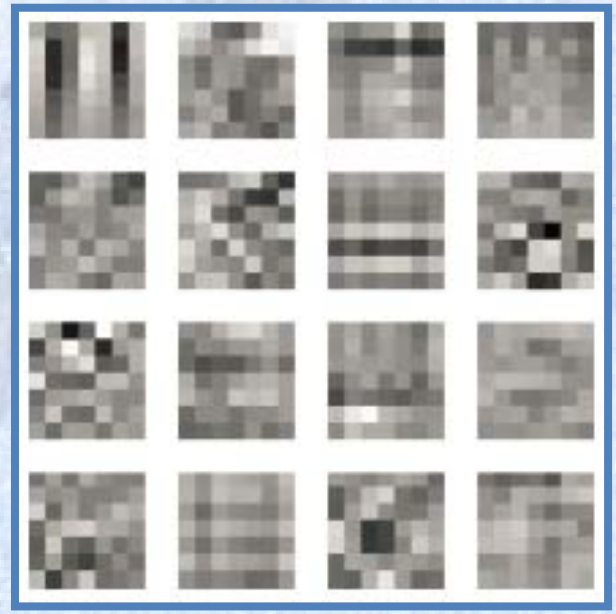


Parameters with random initialization

Kernels are randomly generated from normal distribution with zero mean and a standard deviation(0.8,0.6).

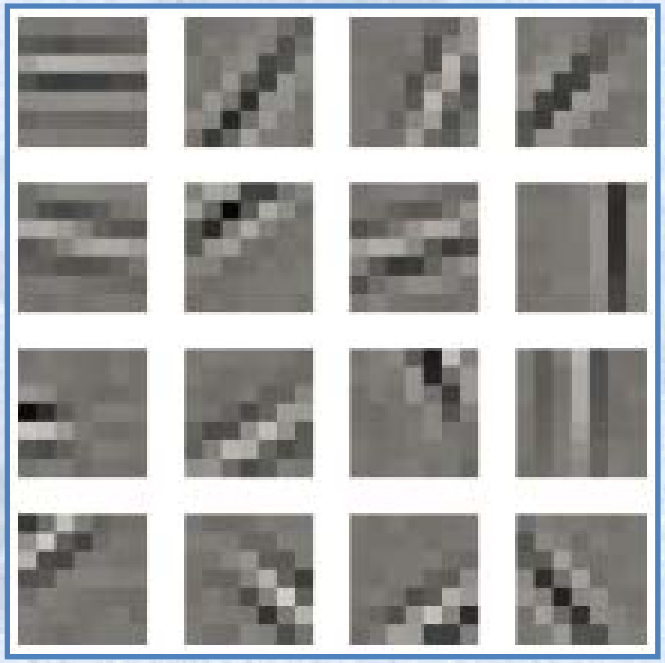
Parameters with supervised learning method

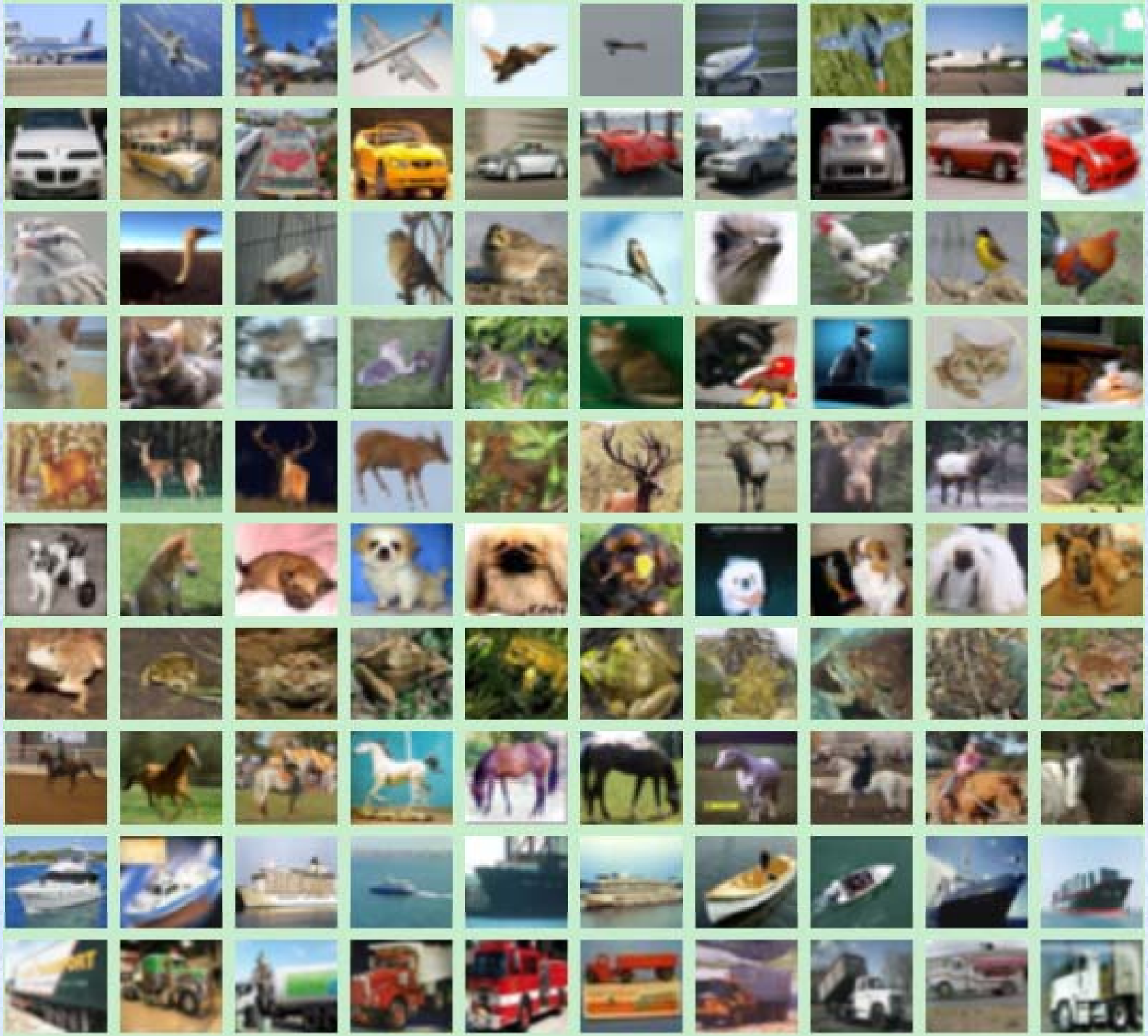
- This network is trained with fully labeled images from the Barcelona dataset.
- Using back-propagation with stochastic gradient descent
- Supervised learning forces kernels to learn common features from the dataset quickly than unsupervised way.



Parameters with unsupervised learning method

- No labeled data is required for training. CIFAR-10 dataset is used to train the network.
- Using K-means clustering learning method.
- The advantage of K-means is that it is fast and biologically plausible than SGD.

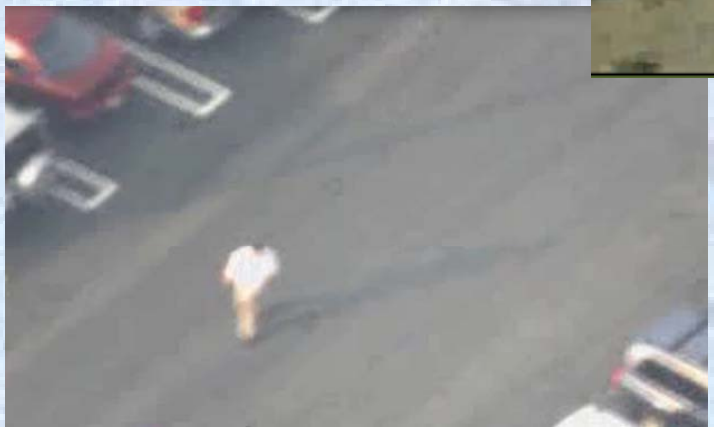




Experimental Results

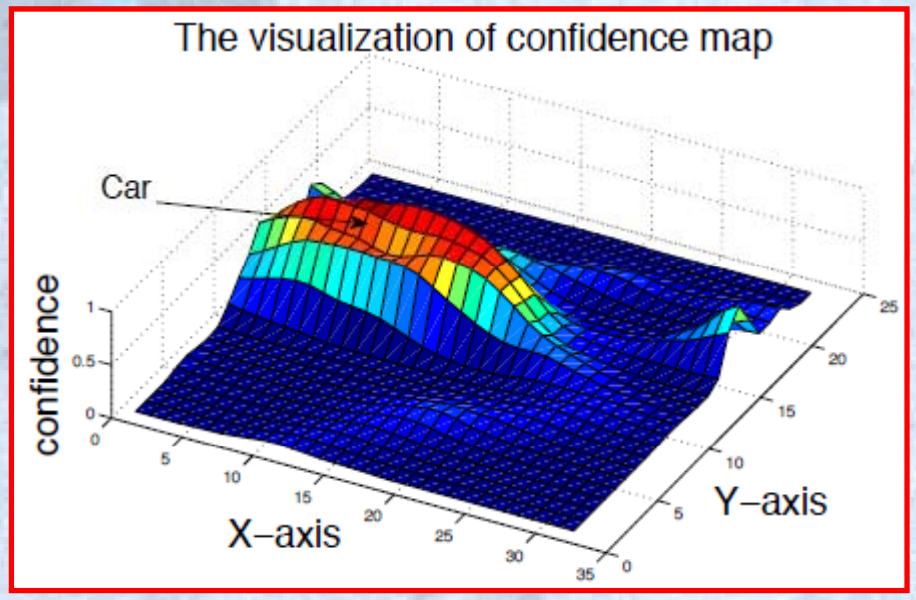
- Testing benchmark: TLD dataset
- Measurement: F = PR/(P+R), P:Precision, R:Recall

Sequence	Frames	ConvNet with random parameters (No pre-learning) Precision/Recall/F-measure	Supervised ConvNet (back-propagation with SGD) Precision/Recall/F-measure	Unsupervised ConvNet (K-means clustering learning) Precision/Recall/F-measure
David	761	0.08 / 0.05 / 0.06	**0.12 / 0.07 / 0.09**	0.08 / 0.05 / 0.06
Jumping	313	0.28 / 0.28 / 0.28	**0.51 / 0.51 / 0.51**	0.22 / 0.22 / 0.22
Pedestrian 1	140	**0.81 / 0.81 / 0.81**	**0.81 / 0.81 / 0.81**	0.64 / 0.64 / 0.64
Pedestrian 2	338	0.36 / 0.46 / 0.41	0.35 / 0.45 / 0.39	**0.61 / 0.64 / 0.63**
Pedestrian 3	184	0.41 / 0.49 / 0.45	**0.48 / 0.57 / 0.52**	0.47 / 0.56 / 0.51
Car	945	0.68 / 0.73 / 0.70	0.37 / 0.40 / 0.39	**0.97 / 0.96 / 0.97**
Motocross	2665	**0.14 / 0.26 / 0.18**	0.12 / 0.23 / 0.16	**0.14 / 0.26 / 0.18**
Carchase	9928	0.20 / 0.21 / 0.21	0.25 / 0.26 / 0.25	**0.38 / 0.43 / 0.40**
Panda	3000	**0.40 / 0.44 / 0.41**	0.36 / 0.40 / 0.38	0.26 / 0.29 / 0.28
Mean	18274	0.26 / 0.29 / 0.27	0.26 / 0.29 / 0.28	**0.35 / 0.39 / 0.37**



Sequence	Frames	ConvNet with random parameters (No pre-learning) Precision/Recall/F-measure	Supervised ConvNet (back-propagation with SGD) Precision/Recall/F-measure	Unsupervised ConvNet (K-means clustering learning) Precision/Recall/F-measure
David	761	0.08 / 0.05 / 0.06	**0.12 / 0.07 / 0.09**	0.08 / 0.05 / 0.06
Jumping	313	0.28 / 0.28 / 0.28	**0.51 / 0.51 / 0.51**	0.22 / 0.22 / 0.22
Pedestrian 1	140	**0.81 / 0.81 / 0.81**	**0.81 / 0.81 / 0.81**	0.64 / 0.64 / 0.64
Pedestrian 2	338	0.36 / 0.46 / 0.41	0.35 / 0.45 / 0.39	**0.61 / 0.64 / 0.63**
Pedestrian 3	184	0.41 / 0.49 / 0.45	**0.48 / 0.57 / 0.52**	0.47 / 0.56 / 0.51
Car	945	0.68 / 0.73 / 0.70	0.37 / 0.40 / 0.39	**0.97 / 0.96 / 0.97**
Motocross	2665	**0.14 / 0.26 / 0.18**	0.12 / 0.23 / 0.16	**0.14 / 0.26 / 0.18**
Carchase	9928	0.20 / 0.21 / 0.21	0.25 / 0.26 / 0.25	**0.38 / 0.43 / 0.40**
Panda	3000	**0.40 / 0.44 / 0.41**	0.36 / 0.40 / 0.38	0.26 / 0.29 / 0.28
Mean	18274	0.26 / 0.29 / 0.27	0.26 / 0.29 / 0.28	**0.35 / 0.39 / 0.37**

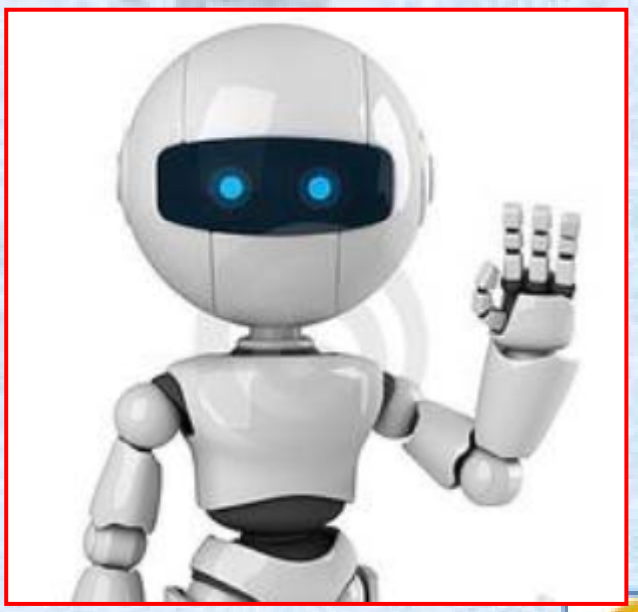
- Perform best in the “Car” dataset.
- Network with K-means performs best.
- It is interesting that random network shows the best records for some datasets.
- It takes about 0.074s seconds to process one frame(13.5 fps) which is 320 ×240 pixel image on a 2-core Intel i-7 laptop.
- Overall performance fluctuates and not comparable as the best trackers such as TLD.


The visualization of confidence map
Car
confidence
1
0.5
0
0
5
10
15
20
25
30
35
X–axis
0
5
10
15
20
25
Y–axis

Analysis and Limitation

- Many groups in machine learning focus on huge networks with sophisticated learning techniques, but most of them cannot be applied to real-time application, e.g. robotic vision.

Huge Network



Analysis and Limitation

- A major drawback of many feature learning systems is their complexity and expense.
- How to improve for real-time applications?
  Simpler algorithm, Feature numbers, Stride step, Reception size……

Learning rate

Weight decay **Huge Network** ……

momentum Sparsity penalties

Analysis and Limitation

- Supervised network is not suitable for a general purpose system such as tracking an unknown object.

  The reason for the failure is that all the parameters in the network are adjusted to increase confidence on the specific categories of the labeled dataset and in the process loses information not relevant to that task.

Analysis and Limitation

- The deep neural network with random kernels in the experiment shows comparable performance to the result of supervised network.
- The results from randomness support the fact that the random kernels themselves can be considered as a set of basic blocks and they are able to extract features in spite of unknown random patterns.

Sequence	Frames	ConvNet with random parameters (No pre-learning) Precision/Recall/F-measure	Supervised ConvNet (back-propagation with SGD) Precision/Recall/F-measure	Unsupervised ConvNet (K-means clustering learning) Precision/Recall/F-measure
David	761	0.08 / 0.05 / 0.06	**0.12 / 0.07 / 0.09**	0.08 / 0.05 / 0.06
Jumping	313	0.28 / 0.28 / 0.28	**0.51 / 0.51 / 0.51**	0.22 / 0.22 / 0.22
Pedestrian 1	140	**0.81 / 0.81 / 0.81**	**0.81 / 0.81 / 0.81**	0.64 / 0.64 / 0.64
Pedestrian 2	338	0.36 / 0.46 / 0.41	0.35 / 0.45 / 0.39	**0.61 / 0.64 / 0.63**
Pedestrian 3	184	0.41 / 0.49 / 0.45	**0.48 / 0.57 / 0.52**	0.47 / 0.56 / 0.51
Car	945	0.68 / 0.73 / 0.70	0.37 / 0.40 / 0.39	**0.97 / 0.96 / 0.97**
Motocross	2665	**0.14 / 0.26 / 0.18**	0.12 / 0.23 / 0.16	**0.14 / 0.26 / 0.18**
Carchase	9928	0.20 / 0.21 / 0.21	0.25 / 0.26 / 0.25	**0.38 / 0.43 / 0.40**
Panda	3000	**0.40 / 0.44 / 0.41**	0.36 / 0.40 / 0.38	0.26 / 0.29 / 0.28
Mean	18274	0.26 / 0.29 / 0.27	0.26 / 0.29 / 0.28	**0.35 / 0.39 / 0.37**

Conclusions

- The experiment has shown that deep neural networks are able to track the target though performance fluctuates depending on the training method.

- Training a network in a strongly supervised way does not guarantee high performance for a general purpose system like tracking an unknown object.

References

- Jonghoon Jin, Dundar, A. , Bates, J. , Farabet, C. , Culurciello, E. Tracking with Deep Neural Networks. Information Sciences and Systems (CISS), 2013.
- Convolution networks and applications in vision. Y. Lecun, K. Kavukvuoglu and C. Farabet. International Symposium on Circuits and Systems. 2010.

Thank you