DE GRUYTER
OPEN

JAISCR, 2017, Vol. 7, No. 2, pp.125– 136
10.1515/jaiscr-2017-0009

A NOVEL DEEP NEURAL NETWORK THAT USES SPACE-TIME FEATURES FOR TRACKING AND RECOGNIZING A MOVING OBJECT

Oscar Chang[1], Patricia Constante[2], Andrés Gordon[2], Marco Singaña [2]

[1]*Department of Software Innovation, Farmaenlace, Quito Ecuador*

[2]*Department of Energy and Mechanics, Universidad de las Fuerzas Armadas*
ESPE Latacunga Ecuador

Abstract

This work proposes a deep neural net (DNN) that accomplishes the reliable visual recognition of a chosen object captured with a webcam and moving in a 3D space. Autoencoding and substitutional reality are used to train a shallow net until it achieves zero tracking error in a discrete ambient. This trained individual is set to work in a real world closed loop system where images coming from a webcam produce displacement information for a moving region of interest (ROI) inside the own image. This loop gives rise to an emergent tracking behavior which creates a self-maintain flow of compressed space-time data. Next, short term memory elements are set to play a key role by creating new representations in terms of a space-time matrix. The obtained representations are delivery as input to a second shallow network which acts as ”recognizer”. A noise balanced learning method is used to fast train the recognizer with real-world images, giving rise to a simple and yet powerful robotic eye, with a slender neural processor that vigorously tracks and recognizes the chosen object. The system has been tested with real images in real time.

Keywords: deep architectures, deep learning, artificial vision.

1 Introduction

In the computer vision world the reliable recognition of a chosen object is a primary necessity and several deep neural nets (DNN) solutions have been lately presented. In [1] the authors use an algorithm based on bayesian optimization that sequentially proposes candidates regions for an object bounding box and a structured loss that explicitly penalize location inaccuracy. In [2] a formulation which is capable of predicting the bounding boxes of multiple objects in a given image is presented. In [3] the authors propose various combinations of architectures with 1 or 2 stages of feature extraction, random, unsupervised and supervised. All these mentioned approaches utilize feed-forward structures where memory elements are not considered. However in nature, short term memory elements play a key role in the processing of visual cognitive tasks [4,5]. This work develops a methodology to train a DNN, whose functioning includes short term memory elements, making possible the definition of a space-time matrix where other neural structure can detect space time features of one chosen object. The matrix contains highly compressed data orderly released in a first-in / first-out mode, coming from a pre trained autoencoder trained with big unlabeled data. The outcome is a novel 2D representation of the chosen object. We prove that these new representations convey significant amounts of space time features and that a second shallow network can be quickly trained to separate the chosen object from white noise. Noise injection in ANN training has been proposed with different points of view in [6, 7]. We use the method described in [8, 9, 10] where, during real world training, representations of the chosen object are shown as examples and noise as counter examples. We also prove that this later training strategy provides enough ”foreign element” immunity as to rapidly separate the chosen object from uncountable many other real world images. Thanks to the utilized highly compressed data, the resulting six layers neural processor has a slender structure that rapidly learns from real world images to vigorously track and recognize the chosen object. Our final testing ground comprises a physical 3D space, a regular webcam and the captioning of images in real time.

Deep Network

The early components of our methodology were first discussed in [8]. Our new developments take these ideas further and use them to build a robotic eye with the capacity to track and recognize a chosen object captioned with a regular webcam. Our design is oriented toward a ”first front” processor, fast, simple, reliable and devoted to the proficient tracking and recognition of one single object that might play a key role in performance and/or survival. The network utilizes six layers of non-linear neurons organized as two independent ”shallow” networks with three layers each. In the middle of these networks there exists an intermediate matrix of short term memory elements that actively participates in recognition tasks. Our processor thus stacks a total of six operative cascaded layers with a middle matrix of short term memory elements. (Figure 1).

The first shallow network, called N_1, is extensively trained by using substitutional reality and an original autoencoding scheme, where the net being trained achieves zero tracking error capacity for one chosen object, in a discrete tracking ambient. This skilled individual, with a highly specialized hidden neurons weight structures, is set to work inside a real world closed loop control system, where images coming from a webcam produce displacement information for a moving region of interest (ROI) inside the own image. The resulting real time loop gives rise to an emergent tracking behavior which, with the support of short term memory elements, creates new feature and data representations, in terms of compressed information. It turns out that the obtained compressed representations convey plenty of space time features that make possible for others downstream processors to rapidly learn to detect the presence of the chosen object.

A subsequent shallow network N_2 acts as a ”recognizer agent” and is trained with real world images as examples and white noise as counter examples, until it recognizes the chosen object. As a second contribution we prove that the extensive pre training and the resulting highly compressed data, favor learning in the real world and that only few images of the chosen object and white noise are enough to awaken solid recognition capacities. The stacking of N_1 and N_2 completes six operative layers that sustain high data compression and work around a matrix of short term memory elements. Operating with real world images the resulting neural machinery behaves as a simple and yet powerful robotic eye controller. To develop our methodology a helipad figure is chosen as object of study. This item is often used in artificial vision and automatic landing processes [11, 12]. The tracking-recognition problem is divided into two lower level problems in a hierarchical manner, by using independent agents where the quality of being and agent implies the capacity to perform a useful job without external intervention and satisfying the four weak conditions of [13]: Autonomy, Social ability, Reactivity and Pro-activeness.

2 N_1: The offset tracking autoencoder

In classical DNN theory an autoencoder is defined as a ANN where the output layer has equally many nodes as the input layer, and instead of training it to predict some target value y given inputs x, an autoencoder is trained to reconstruct its own inputs x [14]. The training algorithm can be summarized as: Unlabeled data is provided as input and the output of the network tries to reconstruct the input.

For the purposes of this work we define an offset tracking autoencoder (OT Autoencoder) as a shallow (three layers) ANN where the output neurons reproduce, as classifiers, the estimated tracking offset between the center of a moving region of interest (ROI) and the center of the image of one cho-

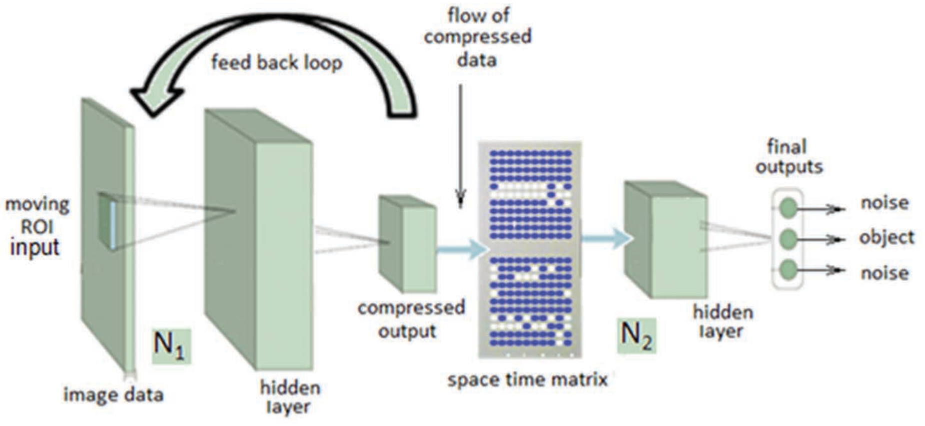


Figure 1. Our deep architecture. A shallow network N_1, pre trained as autoencoder with big data, gains control of a moving region of interest (ROI) inside a bigger image. This incidental loop awakes an emergent tracker agent that produces a spontaneous flow of compressed data. The compressed data is orderly released in a matrix of short term memory elements, creating new 2D representations of one chosen object. This new information is delivery as input to a second shallow network, specialized in cognitive labors and trained with a noise balanced method to distinguish the chosen object representation from the white noise representation. Due to the involved high compressed data the actual net structure has a slender outline that in turn favors low cost and fast processing.

sen object (Figure 2). Notice that net is set to track the existing offset and not the object itself. Since the number of output neurons is finite and they act as classifiers, the measurable offset is discrete and, for this paper, measured in pixels. After training the network will calculate how far is the ROI center from the current image center, regardless of what kind of image is being processed.

The OT autoencoder structure

Figure 1 shows the scheme of the used N_1 autoencoder. For our methodology neurons gain and learning coefficient require specific settings. In this regard output neurons, which behave as on-off classifiers, are assigned a strong 2.5 gain that produces steeply neurons behavior. In contrast hidden neurons, which behave as multi value encoders, are assigned a conservative 0.1 gain. Notice the relation 25/1 in gain setting. To sustain a long learning journey, where millions of unlabeled data elements participate, the learning coefficient was set to a minute 0.01. All biasing elements are assigned a zero value. The net parameters that follow were chosen for convenience and after some trial and error. We maintain that all of them can be freely adapted to other resolutions and hardware capacities.

In the three layers network N_1 the input comprises a ROI retina of 100x100 pixels, with values obtained by applying canny border detection to the original webcam image previously converted from color to gray. This ROI becomes mobile by pairing it to an openCV ROI whose middle coordinates x_r, y_r can be varied under programming control. The hidden layer contains 100 neurons with gain set to 0.1. The output layer contains 22 neurons with gain set to 0.9, grouped into two clusters Cx, Cy with 11 neurons each. Clusters behave as independent classifiers and indicate the net estimation about the offset between x_r, y_r and the helipad center, where the estimation is done with 5 pixels of variation. In normal feed forward operation for each image captured by the webcam the net produces 22 analog values between 0.0 and 1.0. A winner takes all criterion which is applied so that in each cluster the neuron with the highest output is taken as winner. Once found, the two winners point to the estimated Δx_r, Δy_r tracking offsets in pixels. Pixels inside the ROI take the binary analog values 0.1, 0.9 obtained by applying the threshold value 0.5 to the received canny image.

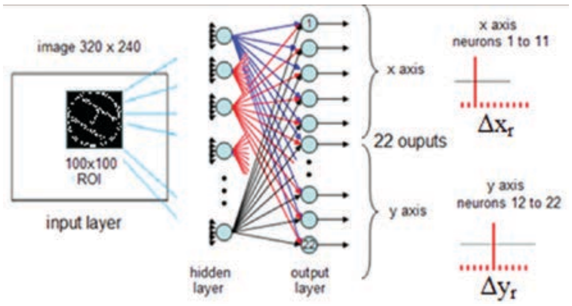


Figure 2. The offset tracking autoencoder N_1. Sigmoidal neurons are organized as a three layers network, where the input vector comprises a two-dimensional region of interest (ROI) of 100x100 pixels moving under programing control inside a bigger 320x240 image. The network outputs act as classifiers and try to calculate the existing tracking offset Δx_r, Δy_r between the center of the ROI and the center of the image, in our case a helipad.

Training of the OT Autoencoder

The training objective is to predict for each training image the tracking offset Δx_r, Δy_r between the ROI center and that helipad center by using the coding principles described in Figure 3. Here the helipad stays fixed and the ROI is displaced in the x and y axis according to two random generated offset values (dx, dy) whose maximal possible values, for the current example, are ±5 pixels, that is each displacement can assume the values: -5,-4,-3,-2,-1,0,1,2,3,4,5. When (dx, dy) take the values (0,0) the ROI center and the helipad center coincide and the 22 targets are settled by raising (set to 0.9) the equivalent cero displacement targets in each axis while setting to 0.1 all other targets (Figure 3-b). For other values the ROI moves off center and the targets assume the corresponding matching values (Figure 3-c to 3-d). The combined offset space has a total of 11x11= 121 elements. As a first training stage the helipad image stays fixed in rotation and scale and only the ROI positon changes. For operative purposes the tracking error is defined as

$$e_g = \sum_{1}^{m}(i[Tx] - j[outx])^2 + \sum_{1}^{m}(i[Ty] - j[outy])^2, \qquad (1)$$

where:

e_g = global error,

$i[T_x]$ = pointer to the only target set to 0.9 in the x axis,

$j[out_x]$ = pointer to the winner in the x axis,

$i[T_y]$ = pointer to the only target set to 0.9 in the y axis,

$j[out_y]$ = pointer to the winner in the x axis,

m = number of outputs.

In pseudo code:

```
  Begin
      Place helipad about the center of the image.
      Place ROI over helipad
      Do
      generate random offset dx,dy in the range
      $-5, +5$;
      set target(x), target(y) according
      to generated offset
      move the ROI to the generated offset;
      forward;
      backpropagation;
until mean error=0 for more than 100 cycles;
end do;
end;
```

As shown in Figure 4, for this first training stage the error decays rapidly and a perfect scored is reached in about 1000 backpro cycles (about 1 minute in common laptop).

Extended training stages

As it will be shown, the operative value of an OT autoencoder depends in its capacity to support wide invariances in position, rotation and scale in the images of the chosen object. To attain this capacity we developed a method where the trainee network interacts with images of the chosen object for a long period using substitutional reality [15] [16], where many different images of the chosen object are computer generated and participate in the training. In harmony with this principle the difficulty of the process to be learned will be made to slowly increase, starting with a fixed object and then adding step by step visual complexities such as gyro and scale changes. This allows time for carefully

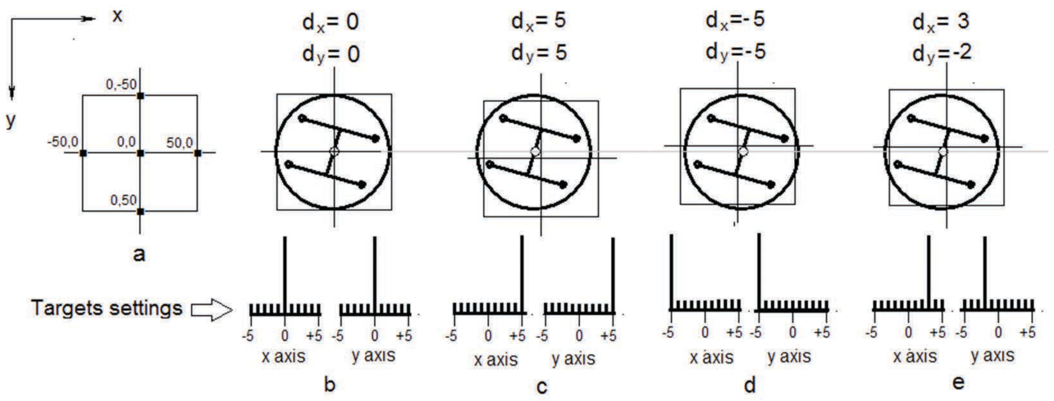


Figure 3. The OT autoencoding strategy. The helipad stays fixed and the ROI is displaced in the *x* and *y* axis according to two random generated offset values dx, dy. a) Map of maximal possible offset displacements. b) The generated offset is (0, 0). The ROI center and the helipad center coincide. Targets are settled by raising (set to 0.9) the equivalent cero offset target in each axis. C) The generated offset is (5, 5). The ROI center moves 5 pixels to the right and 5 pixels down. Targets are raised at the matching locations. d) Offset are -5,-5. ROI center moves up and left. Targets are raised accordingly. e) Offsets are 3,-2. ROI center moves right and up. Targets are raised accordingly.

adjust learning weights, until a minimal long range global error is reached. Our training thus requires a controlled environment where the following conditions are satisfied:

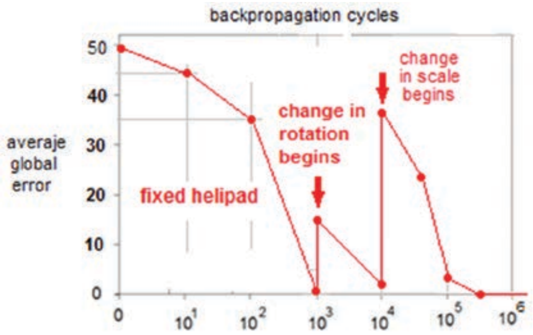


Figure 4. Average tracking error during the extended training stages. Training is done with unlabeled data taken from substitutional reality and during million of backpropagation cycles. For a fixed helipad the error decays rapidly. Given enough time and increasing the problem difficulty by stages the tracking capacity reach the maximum allowed plateau (error =0) and remains there in a stable condition. At this point, at the micro level, no matter what example of the substitutional reality is showed, the tracking error is always zero and supports extensive changes in rotation and scale.

1. There exists a substitutional reality background which generates large data set of unlabeled images of the chosen object (helipad). Images can be handled under programming control and during a prolonged training journey.

2. The degree of difficulty of the training problem increases in successive stages and each stage is completed by using backpropagation.

3. Each stage will be assumed to be completed when a defined tracking error reaches and maintains a zero value during a prolonged period.

This later self-controlled strategy makes possible a long range, unsupervised learning process that automatically advances to upper levels until a highly trained individual is obtained.

Once the Stage 1 above described (fixed helipad) is completed the next following stages are:

Stage 2: Rotation invariance. The helipad does a continuous rotation, one degree per cycle (Figure 5-b).

Stage 3: Scale invariance from 4:3. The helipads changes in scale from 4 (about the size of the ROI) to 3 in 0.01 steps (Figure 5-c).

Stage 4: Scale invariance from 4:2 in 0.01 steps (Figure 5-d).

Stage 5: Scale invariance from 4:1 in 0.01 steps (Figure 5-e).

Notice that during the extended training journey the combined number of possible training images is given by 121 x 360 x 300 = 13.068.000

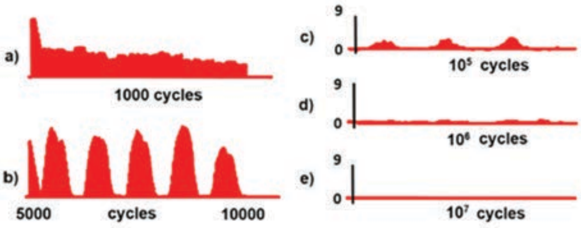


Figure 5. Detailed average tracking error during extended training. a) First stage with helipad fix, error decays rapidly in about 1000 cycles. b) Second stage with rotating helipad, complex cyclic response appears, with peaks caused by gyro c) Error becomes small. d) Stage six, helipad changes in rotation and scale 4:1. Error becomes small and sporadic after millions of backpropagation cycles e) Tracking error disappears.

As shown in Figure 6, after the extended training is completed the hidden neurons weight structures become highly specialized display a harmonic behavior and provide high spatial compression rate: 10.000/22 or 445/1 with a fast two levels operation. These structures have not known helipad recognition capacity by itself and they show no traces of the used helipad.

3 The emergent tracking agent

The trained OT autoencoder obtained in Section 2 is now set to operate in a real world closed loop system, where images coming from a webcam are processed by the autoencoder and used to control the ROI displacement in the next (future) frame, i.e.:

$$x_r(n+1) = x_r(n) - \Delta x_r(n), \qquad (2)$$

$$y_r(n+1) = y_r(n) - \Delta y_r(n), \qquad (3)$$

where:

x_r is ROI position in the x axis,

y_r is ROI position in the y axis,

Δx_r is the x axis offset calculated by the autoencoder,

Δy_r is the y axis offset calculated by the autoencoder.

For operative purposes whenever the ROI borders overpass any image border, $x_r r$ and $y_r r$ will be reset to zero.

In the virtual world and with the trained OT autoencoder in command the calculated Δx and Δy are always error free and as result the system can use this information to move the ROI center to exactly match the helipad center, as long as the current deviations are equal or less than ±5 pixels. In other words if the trained OT autoencoder assumes the ROI movement control, then the moving ROI can perform a zero error tracking job over the helipad. We call this close loop dynamic system a proactive ROI or pROI.

Real world behavior

When the above pROI is brought to the real world and confronted with never seen complex image the resulting behavior is also complex and condensed in the following record:

a Pixel Cluster: When confronted with a never seen dense pixel region (Figure 7-a) the pROI orbits around it in an equilibrium condition. If the dense region moves slowly enough (±5 pixels/frame) the pROI executes a dynamic, spontaneous tracking,

b Unstable region: for some imagery situation, the pROI moves around a complex route (show graph of route) and never settles down,

c Look Alike Element: The pROI proactively centers any image that may look as a helipad. The tracking capacity increases,

d Helipad Image: the pROI centers and tracks the image. The tracking capacity improves.

In resume when handling complex real world images, with uncountable many variations the pROI, driven by a trained autoencoder, behaves as an independent agent whose main behavior rules are:

- With a complex image as input the pROI moves

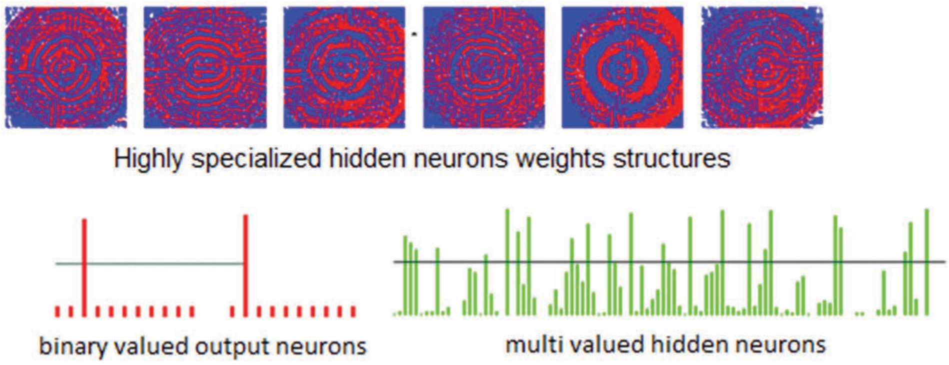


Figure 6. Highly specialized hidden neurons weights structures obtained with the OT autoencoder after extended training. The processing weights distribution assumes harmonic patterns. The system contains 100 of these patterns, none of them equal to each other and none of them showing traces of the original helipad. Output neurons with high gain behave as ideal binary elements (left). Hidden neurons with low gain behave as multi valued coding elements (right).

to an equilibrium point, where a specific cluster of pixels remains centered (trapped),

- After equilibrium the pROI tracks anything that has been trapped, with certain preferences for specific cluster of pixels,
- With white noise and other complex scenario as input the pROI moves randomly with complex trajectories,
- With the helipad as input the pROI improves its tracking capacity.

When handling complex images the pROI generates a self-sustain, continuous flow of compressed information where hidden neurons operate in a multivalued coding mode. The local data compression rate is 10000/22 or 445/1.

The decompressed space time representation

The self-sustained flow of compressed information produced in the previous session has an embedded helipad nature. The following question is: Can the helipad be detected by just watching the compressed output? As an early trial we intended this solution and obtained poor results. This situation was greatly improved by creating a space decompression zone where short term memory elements are organized as a first in first out (FIFO) matrix M given by:

$$M = \begin{bmatrix} e_{1,1}, e_{1,1} & \cdots & e_{1,9}, e_{1,10} \\ \vdots & \ddots & \vdots \\ e_{22,1}, e_{22,1} & \cdots & e_{22,9}, e_{22,10} \end{bmatrix}, \qquad (4)$$

where: e_{ij} is the output value of neuron i in time j.

Notice that 10 time slot have been chosen for this particular case.

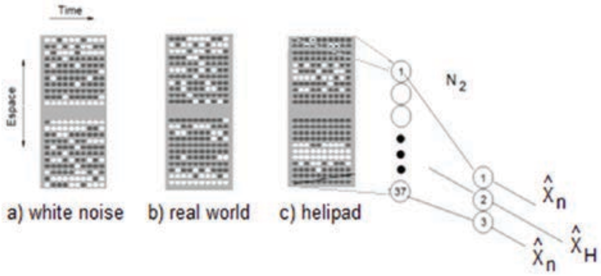


Figure 9. The recognizer network N_2 and the space-time matrix representations for three different elements: a) White noise, b) Typical random selected real world image, c) The helipad. These representations are delivered as input to the shallow network N_2 whose duty is to recognize when the helipad is present in the received images. Due to the utilized highly compressed data, N_2 has a slender structure with 220 input, 37 hidden and three outputs. The central output x_H will indicate the helipad presence. The other two outputs xn are used to generate complementary error values during training.

For each received image frame the calculated output vector enters in the left most columns of the matrix, while all others stored vector values are shifted to the right following a FIFO strategy. The right most stored vector valued is discarded. The obtained space-time matrix is shown in Figure 8. This novel representation will be delivered as input to other trainable network that will act as rec-

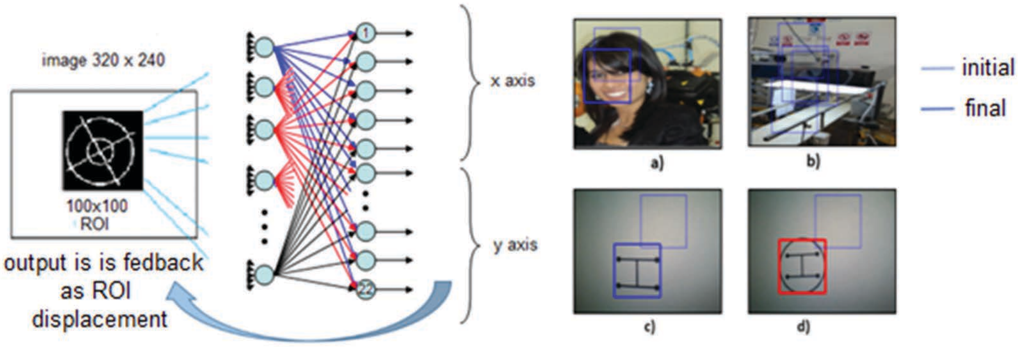


Figure 7. The emergent tracker behavior of the pROI when confronted with never seen real world objects and backgrounds. a) The pROI moves toward a cluster of pixels (face) and orbits around it. If cluster moves slowly enough the pROI executes a dynamic, spontaneous tracking. b) For some complex images the pROI moves around a complex route and never settles down. c) For a look alike element the pROI centers the image and becomes stable. d) For the helipad pROI centers the image and becomes stable.

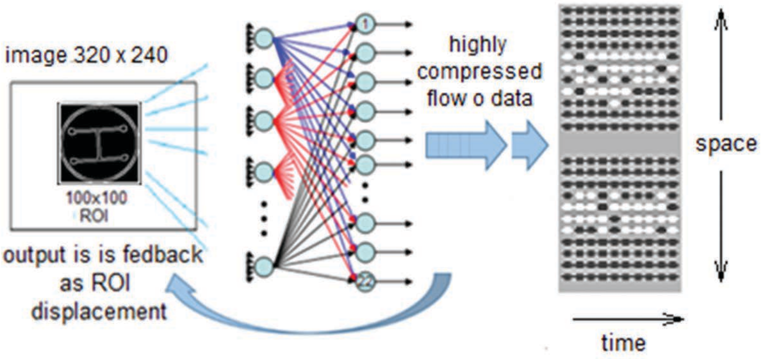


Figure 8. New space-time data representation. A matrix of 22x10 short term memory elements expands in space the compressed data produced by the emergent tracker. Using a common laptop (I5, 3.6 Giga) the system runs at 30 frames/second. This new matrix representation will be fed as input to other participant recognizer network.

ognizer. This matrix changes for each processed frame, in a common laptop (I5, 3.6 Giga) the system runs at about 30 frames/second so the whole matrix content is refreshed each 1/3 of a second.

Table 1. Results: Experiment 1

	Number of Samples	False Helipads
Accordion	20	0
Airplanes	20	0
Anchor	20	0
Ant	17	0
Background google	22	0
Barrel	20	0
Bass	20	0
Beaver	20	0
Binocularr	20	0
Bonsai	21	0
Brain	12	0
Bronstosaurus	18	0
Buddha	17	0
Buterfly	25	0
Chandelier	20	0
Helicopter	15	0
Pizza	22	0
Soccer ball	17	0
Starfish	20	0
Umbrella	19	0
Warch	20	0
Wheelchair	10	0

The recognizer network N_2 The matrix representation in Figure 8 is used to train a second shallow network which utilizes the matrix as example and noise as counter example. The idea is to create an indirect method that quickly and reliably detects the presence of the chosen object. The use of white noise as counterexample allows a speedy training in the real world by just watching a few images of the helipad. The recognizer network N_2 has 220 inputs, 37 hidden and three outputs (Figure 9). The learning coefficient is set to a global 0.25 while all Gains are set to 0.5.

4 The noise balanced training

Our next goal is to train the recognizer N_2 so that its central output neuron learns to fire whenever helipad images are captured by the webcam and not to fire for any other real world image. To achieve a quick training real world helipad images are used as examples and noise as counter examples (Figure 10). We prove that since all space-time matrix representation of the helipad look alike, a few examples of any helipad images are enough to rapidly awaken a large inference capacity about helipads. This is so because the helipad knowledge comes pre-assembled in the highly specialized hidden neurons weight structures shown in Figure 6, and the real world training only adapts the system to possible environmental variations such as used webcam, computer speed, etc.

5 Results

The obtained six layer neural processor has a slender deep structure with data compressions rates of 10000 /22 in N_1 and 220/1 in N_2. After pre training the autoencoder N_1 for about 6 hours, the typical real time training of N_2 occurs in about 8 repetitions of (50+50) cycles of backpropagation, which for a common laptop represents about 12 seconds. The noise balanced training of N_2 represents an efficient way to rapidly put to work the useful processing information stored in N_1, which in turns awakens a high inference capacity. The overall stem becomes an acute eye which successfully rejects look-alike figures such as letters H, P, $\Theta, \varnothing, \Phi$.

Experiment 1. The system is set to randomly explore the Caltech database searching for helipads. Each image is loaded and the pROI moves freely inside it for about 100 frames. As seen in Table 1 in about 4 hours of search no false responses were found.

Experiment 2. The system is set to explore different images in the real world. It recognizes the chosen helipad under variable visual conditions while rejecting look-alike elements and uncountable many others objects and backgrounds. (Figure 11). See the video: https://www.youtube.com/watch?v=JzV3Szbkt0M&feature=youtu.be

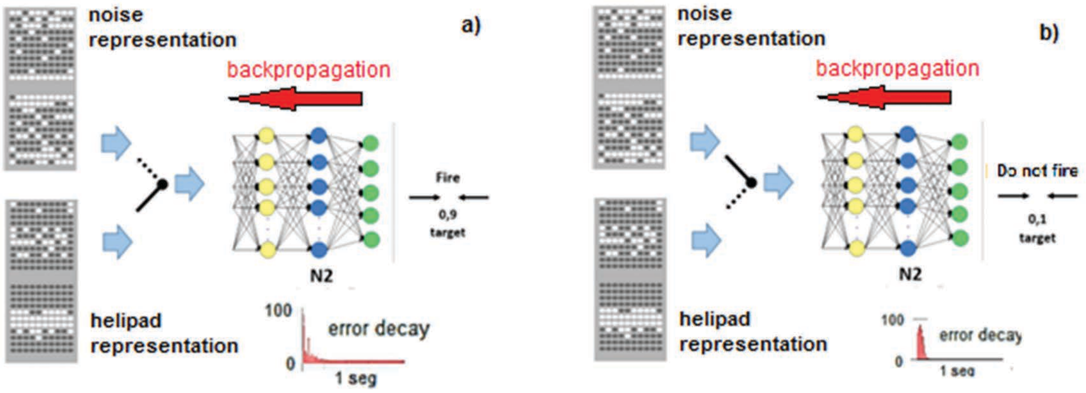


Figure 10. The noise balanced training process. a) Rule 1: The helipad space-time representation is presented as input, .50 cycles of backpropagation follow. During this time the matrix changes dynamically. b) Rule 2: Dynamic white noise is presented as input, the matrix changes dynamically. 50 cycles of backpropagation follow.

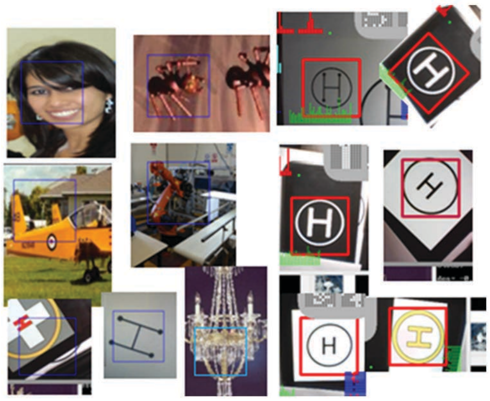


Figure 11. Experiment 2. Some selected rejection and detection results.

6 Discussion

Throughout many practical experiments the only found way to push the OT autoencoder to the zero error plateau is by using the described progressive learning method, starting with a fixed size and releasing rotation and scale variations step by step. According to the obtained results the proposed space matrix representation contains enough feature information as to allow accurate helipad recognition, against dynamic white noise and against countless many other images of the real world.

The training of the recognizer network N_2 involves processing real world images with real world timing, where this later parameter given by the local hardware clock. In our experiments any changes in timing due to hardware characteristics were absorbed by the training itself.

During its entire learning life the deep network only sees the chosen object (helipad) as a hyper example and noise as counter example. This isolated condition defines a strong object-network bond that makes possible a sturdy recognition capacity supporting intricate 3D changes in position tilt and scale. Also by just watching noise as counter examples the processor successfully rejects uncountable many other images coming from a complex real world. We prove that a particular factor learned from configurations of other factors can generalize well.

7 Conclusion

In this work we propose a novel DNN architecture for tracking and recognizing a chosen object whose image is captured by a regular webcam. We introduce an offset tracking autoencoder (OT autoencoder) that is extensively trained with substitutional reality as to achieve zero tracking error in a discrete ambient. The trained autoencoder participates in a real world vision controlled loop and produces a self-motivated tracking agent, capable of predicting locations in never seen images and coding this information in highly compressed data. We enrich the deep model by introducing a matrix of short term memory elements that help to expand the compressed data in space, creating new space-time matrix representations. These elements are delivered as input to a second shallow network which acts as ”recognizer” and is fast trained with real-world images and white noise. The final result is a slender and yet powerful robotic eye controller that operates with high data compression and vigorously tracks and recognizes the chosen object under generic situations.

References

[1] Y. Zhang, K. Sohn, R. Villegas, P. Gang and L. Honglak, Improving Object Detection with Deep Convolutional Networks via Bayesian Optimization and Structured Prediction, Journal CoRR, vol abs/1504.03293, 2015

[2] C. Szegedy, A. Toshev, and D. Erhan, Deep Neural Networks for Object Detection, Advances in Neural Information Processing Systems 26, 2013, pp. 2553–2561

[3] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, What is the Best Multi-Stage Architecture for Object Recognition?, in ’ICCV’, IEEE, 2009, pp. 2146–2153

[4] S. Luck, Visual short term memory, in Scholarpedia, 2007, VOLUME 2, number 6, pp. 3328

[5] E. Averbach, and A. S. Coriell, Short-Term Memory in Vision, The Bell System Technical Journal, 1961, vol. 40, Issue 1, pp. 309–328

[6] Y. Jiang, R. M. Zur, L. L. Pesce and a. K. Drukker, A study of the effect of noise injection on the training of artificial neural networks, in Neural Network IJCNN International Joint Conference on, IEEE, 2009, pp. 1428–1432

[7] Y. Jiang R., M. Zur, L. L. Pesce and K. Drukker, Noise injection for training artificial neural networks: A comparison with weight decay and early stopping, in Medical Physics, vol. 36, 2009, pp. 4810–4818

[8] O. Chang, Reliable object recognition by using cooperative neural agents, in 2014 International Joint Conference on Neural Networks, (IJCNN), Beijing, China, July 6–11, 2014, pp. 2571–2578

[9] Chang Oscar, A Bio-Inspired Robot with Visual Perception of Affordances, In: Computer Vision - ECCV 2014 Workshops: Zurich, Switzerland, September 6–7 and 12, 2014, Proceedings, Part II, editors Agapito, Lourdes and Bronstein, M. Michael and Rother, Carsten, Springer International Publishing. Switzerland, 2015, pp. 420–426

[10] P. Constante, A. Gordon, O. Chang, E. Pruna, I. Escobar, F. Acua, Artificial Vision Techniques for Strawberrys Industrial Classification, IEEE Latin America Transactions, 2016, In-press

[11] M. H. Mahoor, R. Godzdanker, K. Dalamagkidis and K. P. Valavanis, Vision-Based Landing of Light Weight Unmanned Helicopters on a Smart Landing Platform, in Journal of Intelligent Robotic Systems, vol. 61, 2011, pp. 251–265

[12] I. F. Mondragon, P. Campoy, C. Martinez and M. A. Olivares-Mendez, 3D pose estimation based on planar object tracking for UAVs control, in Robotics and Automation (ICRA), 2010 IEEE International Conference on, 2010, pp. 35–41

[13] M. Woolridge and N. R. Jennings, Intelligent agents: Theory and practice, in: The Knowledge Engineering Review, vol. 10, no. 2, 1995, pp. 115–152

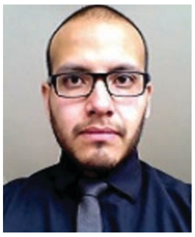
[14] Y. Bengio, Learning Deep Architectures for AI, In: Found. Trends Mach. Learn, vol. 2, number 1, 2009 pp. 1–127

[15] K. Suzuki, S. Wakisaka, and N. Fujii, Substitutional reality system: a novel experimental platform for experiencing alternative reality, In: Scientific reports, vol. 2, 2012

[16] A. Dorian, Can we build a conscious machine?, In: CoRR, 2014, vol. abs/1411.5224

Oscar Chang received his Electrical Engineering with honors at the National Polytechnic Institute in Venezuela. He completed his M. Sc. and Ph. D. degrees in Electrical Engineering at the Pennsylvania State University, USA. He has been a researcher and IVIC, Institute of Engineering, full professor at Simon Bolivar University and visiting professor at the Polytechnic University of Madrid. He is currently an active professor at the UCV in Caracas. His research interests are deep architectures, deep learning, computer vision, artificial intelligence, robotics and bio-inspired devices. He currently works as coordinator of Innovation in the System Department of FARMAENLACE in Ecuador. He is an active member of the IEEE. ogchang@gmail.com

Patricia Constante received her Mechatronics Engineering degree in 2012 at the Army Polytechnic University (ESPE) and her Masters in design, production and industrial automation in 2015 at the National Polytechnic University (EPN), both in The Republic of Ecuador. Her research interests include artificial intelligence, computer vision and industrial automation.



Andres Gordon received his Mechatronics Engineering degree in 2013, at the Army Polytechnic University (ESPE) and his Masters in design, production and industrial automation in 2015 at the National Polytechnic University (EPN), both in The Republic of Ecuador. His research interests include artificial intelligence, computer vision, industrial automation and computational mechanics.

Marco Singaña is an Electronic Engineer with postgraduate studies in industrial digital networks, M.Sc. networks and telecommunications and pursuing the Master's program in design, production and industrial automation at the National Polytechnic School (EPN) in Ecuador. He is currently Professor and Director of the Department of Energy Sciences and Mechanics of the University of the Armed Forces ESPE. His research interests include industrial automation, robotics, mechatronics, computer vision and computational mechanics. Email: masingania@espe.edu.ec.