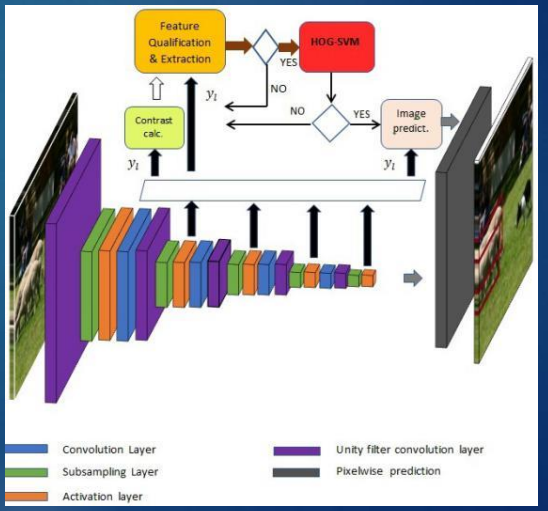
Improving the Performance of Deep CNN: Hystogram Oriented Modelling & Deep Evolutionary-based Learning (DEL)

- ***Automatic detection and recognition*** **of predefined objects in a** ***video stream of Long Range Surveillance Systems*** **is multidimensional problem regarding the** ***complexity of the background, moving sensor(s)*** **and low probability pattern recognition of targets.**
- ***Hystogram Oriented Modelling is incorporated to Deep CNN (DCNN) to reduce the low contrast and noise effect of background to the recognition performance.***
- ***An Evolutionary Strategies Based Optimization while Training algorithm (named as Deep Evolutionary Learning-DEL) is also adapted to train and optimize the overall DCNN.***
- **The proposed topology remarkably improves the** ***detection&recognition performance*** **rate, regardless of the** ***noisy*** **and** ***variable background*** **and the** ***contrast*** **quality.**

THE NETWORK TOPOLOGY

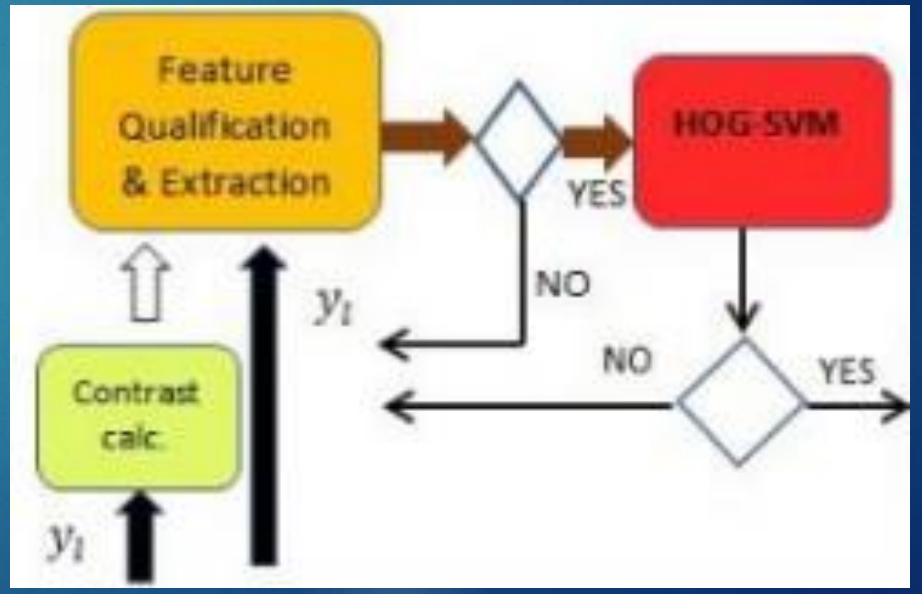
- State of the art Deep Convolutional Neural Networks (DCNNs) are proposed [1-4] which establish both detection & recognition of the objects on a single image frame. Since DCNNs are highly sensitive to the background variations and noise, they have not been used effectively for long range surveillance detection of low contrast objects.
- To overcome the drovebacks the following network topology has been proposed.
- DCNN ınput & hidden layers
- Feature qualification & fast decision (following DCNN activation layers)
  - ✓Histogram Based Contrast Calculation
  - ✓Semantic Feature Qualification & Extraction Structure-FNN
  - ✓HOG-SVM
  - ✓Decision
- Output Layer
  - ✓Image Prediction
  - ✓Pixelwise Prediction



FEATURE QUALIFICATION & FAST DECISION LAYER

- Histogram based Cumulative Contrast Calculation (with parallel processing, once for ten frames, during 40 FPS

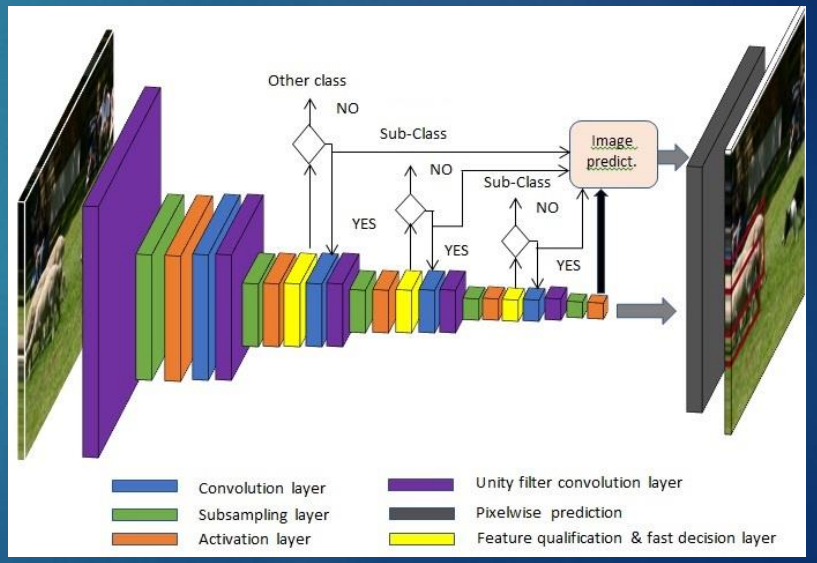
  serveillance)
- Max-Pooling of input to improve performance of overall process
- Semantic Feature Qualification & Extraction Structure-FNN
- Pre-Decision
- Classification HOG-SVM
- Decision



CLASS & SUB-CLASS BEHAVIOUR OF THE TOPOLOGY

- Histogram based Cumulative Contrast Calculation (with parallel processing, once for ten frames, during 40 FPS serveillance)

- Re-interpretition of proposed network topology is shown with figure.
- Feature qualification & fast decision structure acts as Layer of DCNN having activation layer output features (y_l) As input.
- Early and qualitative decisions increase performance.
- Fuzzy decision boundaries for hidden layers, improves flexibility for other class and sub-class decision mechanism.



HISTOGRAM BASED CONTRAST CALCULATION

- Taking activation layer output features as input
- Image devided into 16 equal partition
- Image contrast effect is calculated for each partition as follows:
  - ✓ Cumulative histogram of image is calculated to find real cumulative frequency of image: $freq(y_l)$
  - ✓ Linear regression of cumulative histogram is applied: $RCF(y_l)$
  - ✓ Root mean square error (rmse) of the differences is calculated to find overall contrast effect such that

$$T_l = RMSE(RCF(y_l) - freq(y_l))$$

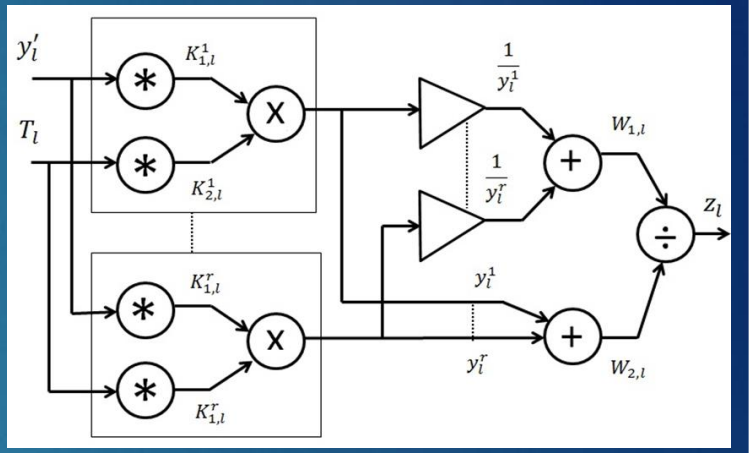
- Calculated with parallel processing, once for ten frames, during 40 fps serveillance.

SEMANTIC FEATURE QUALIFICATION & EXTRACTION STRFUCTURE-FNN

- Fuzzy neural network (FNN) with three layers is used:
  - ✓ Input Layer-Convolution (3x3 Filters as for Mambership Functions)
  - ✓ Fuzzification Layers
    - Hidden Layer
    - Output Layer (Normalized Output
- The membership function in the input layer; convolution layer with 3x3 kernel matrixes forming filters (number of filters = 2xr (rule/feature number)) for inputs:
  - ✓ Max-pooled and normalized input features $y_l^{'}$ and
  - ✓ Image contrast effect T_l.
- With kernal matrixes
- The outputs Y_l^r of the hidden layer can be defined as the followingconvolution functions:

$$Y_l^r = \left(K_{1,l}^r * (T_l, y_l')\right) \times \left(K_{2,l}^r * (T_l, y_l')\right)$$

- Outputs wn,l : $W_{1,l} = \sum_{i,j}^{r}(Y_l^r) \wedge W_{2,l} = \sum_{i,j}^{r}\left(\frac{1}{Y_l^r}\right)$
- The normalized output layer: $Z_l = {}^{W_{2,l}}/_{W_{1,l}}$



TRAINING THE NETWORK TOPOLOGY

- Supervised training is used during the training process both for DCNN and FNN.
- Self adapting evolution based genetic parameters combined with Simulated Annealing (SA) is used for the optimization while training process [5].
- First DCNN is trained/optimized.
- For training of FNN, training input values (y_l) are taken from activation function layer outputs of trained DCNN.
- The optimization algorithm trains variables: *n* x *m* number of kernel matrices $K_{n,l}$ (for each *m* number of layers output (y_l) of *n* number of features) while minimizing the overall error function $e(y)$:

$$E(x) = \sum_{l=1}^{m} \left[w_{e_l} \times \left(\frac{f_l(y_{l-1}) - f_l^d(y_{l-1})}{2\lambda_l (f_l(y_{l-1})_{max} - f_l(y_{l-1})_{min})} \right)^2 \right] + g_0$$

- Where f_l is current layer function and f_l^d is the desired function value. The constraint g_0 acts as bias for better convergance to local minama. The scalar weight can be adapted between [0,1] to have a better solution. An adapted form of ES type (μ, λ)-es algorithm is used as the core of optimization algorithm and for the self adaptation of the step sizes.

TRAINING THE NETWORK TOPOLOGY

Self adaptation of the step sizes:

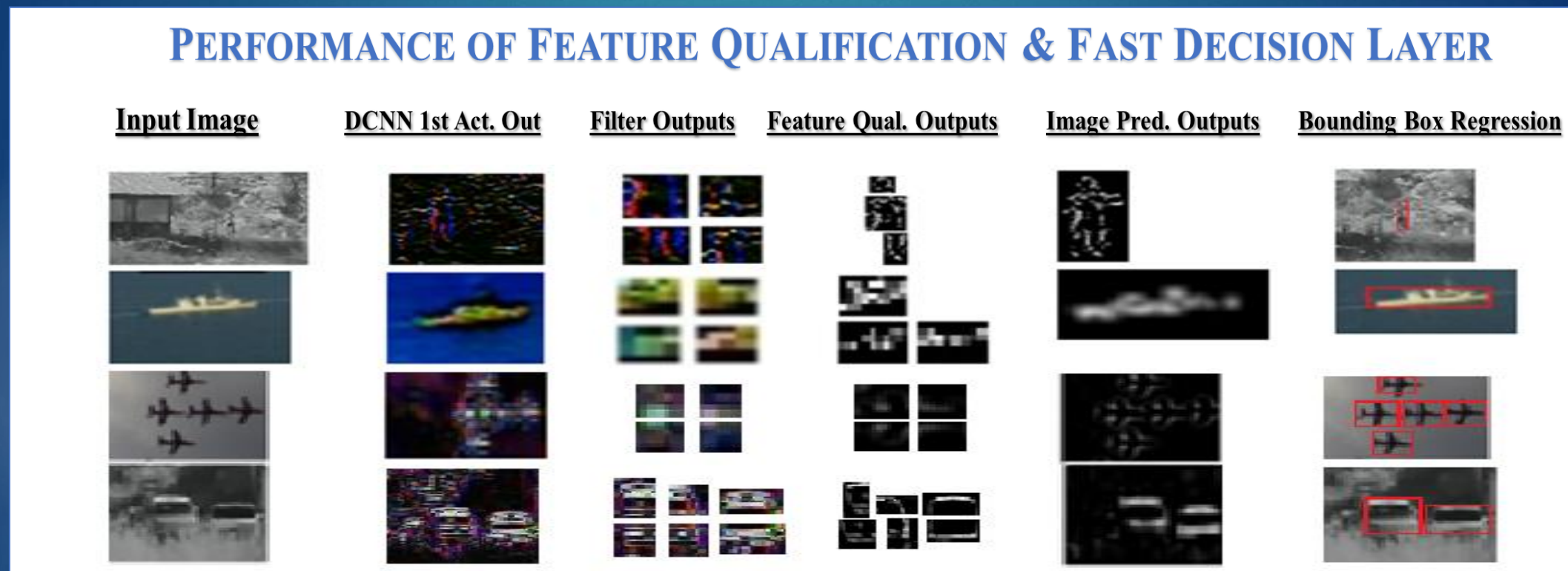
- Self adaptation of the parameters via mutation-recombination of ES, related to annealing temperature of SA, forms the bases of trainer together with error function $e(y)$.
- To increase the overall performance of the ES trainer, the recombination and mutation processes of the (μ, λ)-ES algorithm are utilized to adapt the step length of the object variable parameters within the framework of SA.
- In the self-adaptation of strategy parameters (σ, α), the standard deviation for mutation becomes part of the individual and evolves by mutation and recombination just as the object variables do.
- The strategy parameters are denoted as σ and α and they determine the variance and covariances of the n-dimensional distribution. The variables (of kernel matrices $K\_(n,l)$) are adapted with these strategy parameters and with the selection mechanism, this is called self-adaptation of the step sizes.
- The main loop of (μ, λ)-es algorith is formulated as follows [6]:

$$P^{(m+1)} = sel_{\mu}^{\Lambda}\left(\bigcup_{i=1}^{\Lambda}\{mut(rec(P^{m}))\}\right)$$

- At current annealing temperature t, adaptation of variable x (i.E., (X + Δxλ) where λ is the offspring individuals, cab be derived for ES algorithm as follows (where x stands for variables of $K\_(n,l)$) :

$$(x+\Delta x)_{\Lambda} \equiv \left[P_{\Lambda,\mathrm{N}}(T)\right]' \ni \left[P_{\Lambda,\mathrm{N}}(T)\right]' = \left[mut\left[rec\left[P_{\Lambda,\mathrm{N}}(T)\right]\right]\right]_{\Lambda,n}$$

PERFORMANCE OF FEATURE QUALIFICATION & FAST DECISION LAYER



EXPERIMENTAL RESULST

- The overall DNN topology (DCNN+FNN+SVM) is implemented by using C++ for windows (intel CORE i7) and linux (ARM CORE4) environments.
- The proposed method applied to standard visual object detection tasks on pascal VOC 2012.
- Tests applied on 20 classes of pascal voc 2012 and on 10 classes with continuously searching surveillance system.
- 500 training sample images were taken for training of each classes both from pascal voc 2012 and records from continuously searching surveillance system.
- 500 testing sample images were taken for testing of each classes from pascal voc 2012 and 1000 recorded images from continuously searching surveillance system (having different background and illuminance, contrast levels).
- Taking intersection over union (iou) ≥ 0.7, calculated performances for mean average precision (map) of the proposed topology, DCNN (trained with proposed trainer) and R-CNN (with alexnet and VGG) are given with table 1.
- The comparison curves of the proposed training algorithm to stochastic gradient descent (sgd) and tanh(x) are given with figure 5.
- Map 48.5 for ıou > 0.7 (with STD 0.14) was achieved with continuos serveillance system.
- Eventough much better results were taken comparing r-cnn, for more realistic comparison r-cnn will also be implemented with the same environment.

EXPERIMENTAL RESULST

Model	mAP
R-CNN (AlexNet)	27.1
R-CNN (VGG)	31.3
DCNN	32.1
DCNN+(FNN&SVM)	52.1

Table 1: mAP performances for IoU ≥ 0.7 (PASCAL VOC 2012)

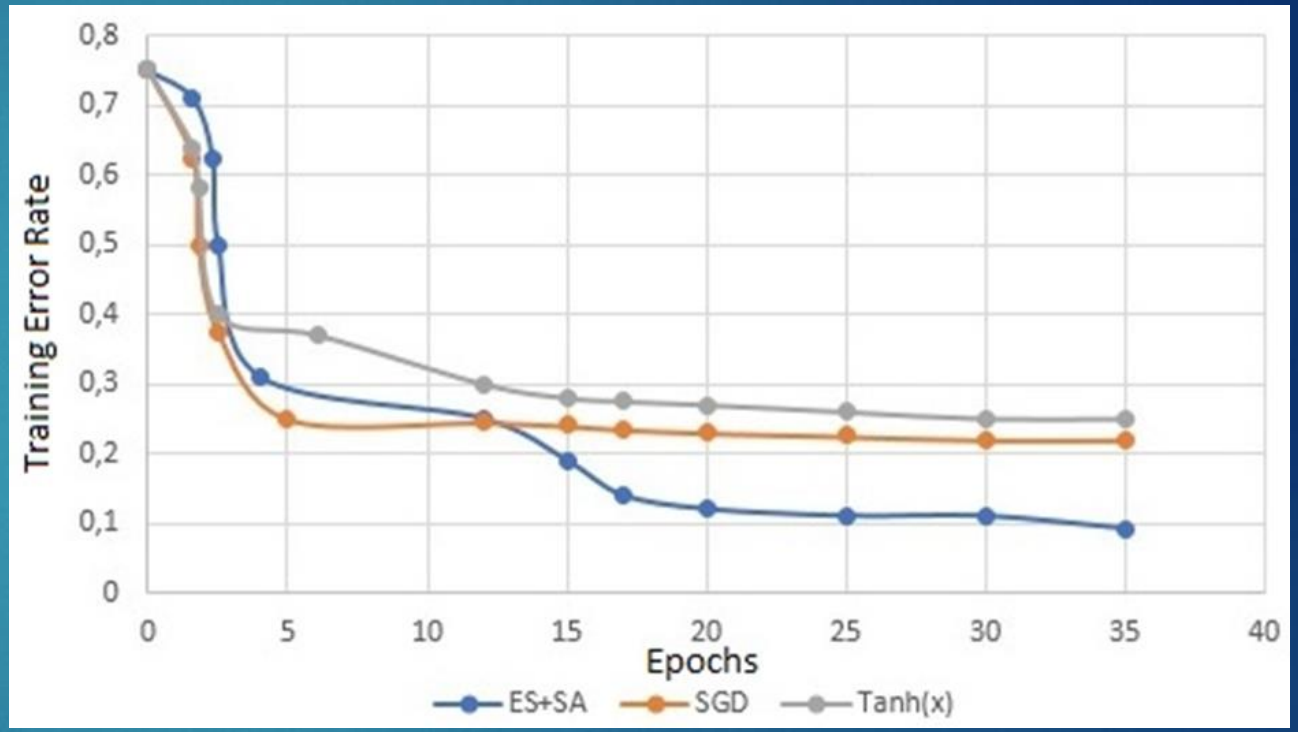


Figure 5: Error Minimization respect to Time consumption (Epochs) of Proposed (DEL) ES+SA, SGD and Tanh(x) training algorithms.

REFERANCES

[1] Long, Jonathan and Shelhamer, Evan and Darrell, Trevor, "*Fully convolutional networks for semantic segmentation", CVPR* 2015.

[2] Y. Zhang and K. Sohn, "Impruving Object Detection with Deep CN via Bayesian Optimization and Structured Prediction", CVPR 2015.

[3] C. Wang, L. Mauch, Z. Guo and B. Yang, "On Semantic Image Segmentation Using Deep Convolutional Neural Network with Shortcuts and Easy Class Extension", IPTA 2016.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks." In NIPS, 2012.

[5] G. Alpaydın, G. Dündar and S. Balkır. "Evolution-Based Design of Neural Fuzzy Networks Using Self-Adapting Genetic Parameters". IEEE Transactions on Fuzzy Systems, Vol. 10, No. 2, April 2002, pp.211-221.

[6] T. Back and H.P. Schwefel, "*Evolution Strategies I: Variants and their computational implementations,*" in Genetic algorithms in Engineering and Computer Science, 1995.