Prediction & Recognition of Fooled Images by Optimizing SVM using PSO

Uttra Singh
Department of Information Technology
NIIST
Bhopal, India
singh.uttra93@gmail.com

Asst. Prof. Angad Singh
Department of Information Technology
NIIST
Bhopal, India
Angada2007@gmail.com

Abstract—Here in this paper an efficient technique for Recognition and classification of Fooled Images using Random Walker Segmentation is done. The Proposed methodology implemented here provides high accuracy rate of Recognizing Fooled Images as compared to the existing technique implemented. The Methodology implemented Support Vector machine is Optimized by Particle Swarm Optimization with Random Walker Segmentation for the Segmentation of Features extracted from the images. The Experimental results are performed on MNIST Datasets and it provides accuracy of 95.67%.

Index Terms—Fooled Images, Deep Neural Network, Support Vector Machine, Particle Swarm Optimization, Random Walker Segmentation.

I. Introduction

In modern years "deep learning" designs exclusively, schemes that approximately imitate the visual or auditory cortex with a objective of bearing out image or video or sound processing jobs have been receiving a lot of interest both in the methodical community and the well-liked media. The awareness this effort has received has fundamentally been acceptable due to the impressive realistic achievements of some of the research involved. In image classification, particularly (the difficulty of recognizing what kind of object is exposed in a picture, or which person's features is shown in a picture), deep learning techniques have been very unbeaten, coming sensibly secure to human performance in different contexts. Modern deep learning systems can be skilled by either supervised or unsupervised techniques, but it's the supervised-learning advances that have been receiving the enormous effects. Recent research in deep networks has significantly improved many aspects of visual recognition [1, 2]. Co-evolution of rich representations, scalable classification methods and large datasets has resulted in many commercial applications [3]. However, a wide range of operational challenges occur while deploying recognition systems in the dynamic and ever-changing authentic world. A huge common of recognition schemes are considered for a standing closed world, where the most important theory is that all categories are known a priori. Deep networks, like many characteristic machine learning devices are calculated to execute closed set recognition. Modern work on open set recognition [4] and open world recognition has make official procedures for executing recognition in settings that want eliminating unknown objects during analysis. While one can always train with an “other” class for uninteresting classes (known unknowns), it is not viable to train with all promising examples of unknown objects. Consequently they require happens for designing visual recognition devices that formally account for the “unknown unknowns”. A usual approach for opening a deep network is to be appropriate a entrance on the output chance. We think about this as discarding uncertain predictions, rather than rejecting unknown classes. Scheirer et al. [4] defined open space risk as the risk associated with labeling data that is “far” from known training samples. That work provides only a general definition and does not prescribe how to measure distance, nor does it specify the space in which such distance is to be measured. In order to adapt deep networks to handle open set recognition, we must ensure they manage/minimize their open space risk and have the ability to reject unknown inputs.

The problem is that for higher layers, the invariance’s are extremely complex so are poorly captured by a simple quadratic approximation. Our approach, by contrast, provides a non-parametric view of invariance, showing which patterns from the training set activate the feature map. Donahue et al., [5] show visualizations that identify patches within a dataset that are responsible for strong activations at higher layers in the model. Despite this encouraging progress, there is still little insight into the inner operation and performance of these composite representations, or how they accomplish such good presentation. From a technical point of view, this is extremely inadequate. Without comprehensible accepting of how and why they effort the improvement of enhanced forms is decreased to trial-and-error. In this paper here they initiate a visualization method that exposes the input stimuli that motivate individual characteristic maps at any layer in the model. It also permits us to watch the development of elements for the duration of training and to identify potential difficulties with the representation. The visualization method they propose uses a multi-layered De-convolution Network (deconvnet), as proposed by [6] to development the characteristic establishments reverse to the input pixel space. They also present an understanding study of the classifier output by occluding parts of the input image exposing which parts of the picture are significant for classification.

II. Literature survey

Nguyen et al. [7] have investigated a reverse problem. From original image data set, they have created visually meaningless images not recognizable by humans, which are classified by a neural network as one of the classes with confidence reaching 99.99%. The authors named these examples "fooling" images. This problem can be explained by creating a special class for fooling images. Training a network this way make it difficult to find new fooling images, since the network has learned features generic to these fooling images. Nguyen et al. made a hypothesis that these fooling examples are based by the discriminative character of classifier, permitting algorithm to find an example that is far away from discriminative boundary with from all the data that has been seen before.

Gu & Rigazio [8] used various preprocessing methods to diminish adversarial perturbations. They have tested several denoising procedures including injection of additional Gaussian noise and subsequent Gaussian blurring. More sophisticated methods using autoencoder trained on adversarial examples or standard denoising autoencoder proved to be more effective. Autoencoders could easily learn simple structure of adversarial perturbations in order to eliminate them. Despite the ability of DNN stacked to the top of the autoencoder to handle adversarial perturbations of the original network, the stacked network became more sensitive to new adversarial examples. New adversarial examples required smaller perturbations than adversarial examples of the original network to perturb it. Gu & Rigazio believe DNN's sensitivity is affected by training procedure and objective function rather than by network topology. As a possible solution to achieve local generalization in the input space, they propose a deep contractive neural network.

Mahendran and Vedaldi [9] also showed the importance of incorporating natural-image priors in the optimization process when producing images that mimic an entire-layer’s firing pattern produced by a specific input image. We build on these works and contribute three additional forms of regularization that, when combined, produce more recognizable, optimization-based samples than previous methods. Because the optimization is stochastic, by starting at different random initial images, we can produce a set of optimized images whose variance provides information about the invariance’s learned by the unit.

So far, our discussion of convolution has tended to the abstract, and the reader would be justified to ask “what’s the point?” In fact, convolutions are capable of transforming images in many useful and concrete ways, like emphasizing edges and computing gradients of hue and value. Moreover, deep successions of convolutions have been shown to produce image encodings that are favorable for classification, namely due to invariance to translation and deformation [10]. But exactly what is computed—and its usefulness for classification—depends on the filters used, and therefore success of a convolutional network crucially depends crucially on choosing good filters.

But to architect a deep network is not insignificant. There are three main challenges. First, because convolutional networks compose many functional components—whose importances as entities and as and entire are not well understand [11] they are complicated to propose. Second, each part of a network may have dozens of hyper-parameters related with it, all of which must be adjusted for peak concert. And finally, the complication of neural networks has confined them from the precise formalism of other areas of machine learning, so practitioners can only rely on unreliable effects to show design. An intimate community has listening carefully on convolutional networks, so a small amount of identify of good design heuristics. Additionally, the common of this community is currently engaged by Google and Facebook. Given these challenges, how can a comparative beginner to field become skilled at to acquire results on equivalence with the field’s leading experts?

A recent study [12] revealed that changing an image (e.g. of a lion) in a way imperceptible to humans can cause a DNN to label the image as something else entirely (e.g. mislabeling a lion a library). Here we give you an idea about a associated result: it is simple to create images that are entirely unrecognizable to humans but that modern DNNs believe to be recognizable purposes with 99.99% confidence (e.g. labeling with certainty that white noise static is a lion). in particular, they take convolutional neural networks trained to present well on both the ImageNet and MNIST datasets and then get images with evolutionary algorithms or gradient ascent that DNNs label with elevated confidence as be in the right placing to each dataset class. It is achievable to create images entirely unrecognizable to human eyes that DNNs believe with near confidence are recognizable entities, which we identify “fooling images”. Our outcomes shed light on importance differences between human vision and existing DNNs and increase questions about the generality of DNN computer vision.

Currently, the most successful models for visual recognition are the deep neural networks (DNNs) [13]. DNNs are neural networks consisting of several layers. Their depth enables them to learn deeper representations of data leading to overwhelming performance over other machine learning methods. Over the past few years, DNNs have gained a lot of interest by researchers as well as by the industry. However DNNs were designed in early 80s, there were insufficient computational resources and knowledge how to train such networks. Training of deep neural networks proposed in early 80s became feasible the recent years, with the vast improvement in computational performance, resulting in shorter training time. Large databases of images essential for training DNNs became available due to enhancement of communication availability and bandwidth. The state of the art deep neural networks show remarkable results in complex tasks such as image classification and speech recognition [13].

III. Proposed Methodology

The Proposed Methodology implemented here consists of Following Stages for the Recognition and Classification of Fooled Images.

1. Take an input MNIST Training Dataset.
2. Apply Support Vector Machine on the Training Input Dataset.
3. Optimize the trained features of Support vector Machine using Particle Optimization.
4. Apply Random Walker Segmentation on the extracted features using PSO-SVM to extract the detected fooled Images.
5. Classify and recognize the final fooled images.

Support Vector Machine

Consider training sample$\{(x_i, d_i)\}$, where x_i is the input pattern, d_i is the desired output:

$$W_0^T X_i + b_0 \geq +1, for\ d_i = +1$$

$$W_0^T X_i + b_0 \leq -1, for\ d_i = -1$$

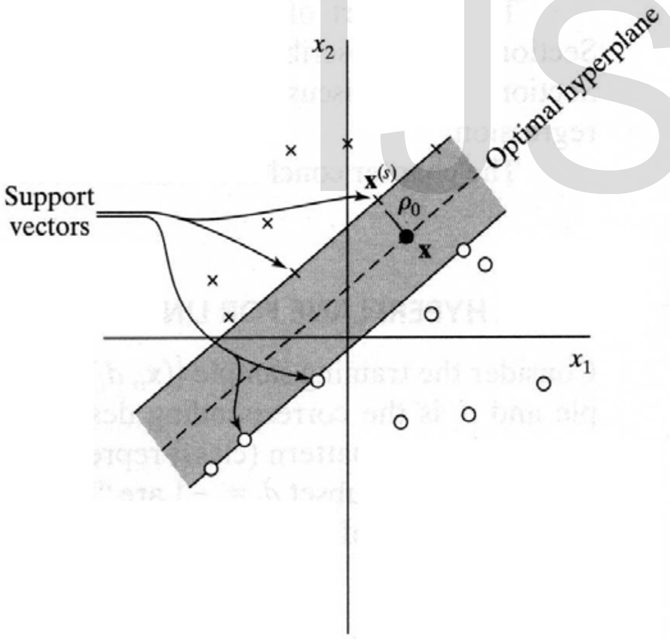

Figure 1 Basic Architecture of SVM

The data point which is very near is called the margin of separation ρ

The main aim of using the SVM is to find the particular hyperplane of which the margin ρ is maximized

Optimal hyperplane $W_0^T X + b_0 = 0$

For example, if we are choosing our model from the set of hyperplanes in *Rn*, then we have:

$$f(x; \{w; b\}) = \mathrm{sign}(w \,.\, x + b)$$

We can try to learn *f*(*x;* _) by choosing a function that performs well on training data:

$$R_{emp}(\alpha) = \frac{1}{m}\sum_{i=1}^{m} l(f(x_i, \alpha), y_i)$$

Particle Swarm Optimization

To overcome the above issues as discussed and for providing efficient Recognition of Unrecognizable Images Some Optimization algorithm such as “Particle Swarm Optimization” technique is used for the optimization of Training Rate.

PSO is a global optimization approach, and was firstly discussed in 1995 by Dr. James Kennedy and Dr. Russell Eberhart. This method was inspired by social behaviors of animals and biological populations. In fact, it is a simulation of a simplified social model like bird flocking and fish schooling. PSO was originally an optimization method for continuous nonlinear functions, i.e., the search space is continuous and decision variables are encoded into real numbers. However, several discrete versions of the algorithm have been proposed in literature. In PSO, there is a population of finite individuals which are called particles. Some advantages of PSO in comparison to other heuristic search algorithms such as GA are ease of its implementation, its fewer parameters for adjustment, its fewer operators and high rate of its convergence. Particle Swarm Optimization (PSO) is easier to implement and it is easy the parameters of PSO. Particle Swarm Optimization (PSO) is also used for maintaining the variety of swarm. Basic PSO has been designed in two steps, i.e., randomly initializing a population, and iteratively updating velocities and positions.

PSO has been utilized in many areas that uses the soft-computing approaches, such as training neural networks, optimizing power systems, fuzzy control system, robotics, radio and antenna design and computer games. PSO algorithm is a multi-agent parallel search technique which maintains a swarm of particles and each particle represents a potential solution in the swarm. In this algorithm particles fly through a multidimensional search space where each particle is adjusting its position according to its own experience and that of neighbors. Suppose $x_i(t)$ denote the position vector of particle in the multidimensional search space at time step t , then the position of each particle is updated in the search space by-

$$X_i(k+1) = X_i(k) + V_i(k+1) \qquad (1)$$

The Basic form of Particle Swarm Optimization (PSO) consists of the moving velocity of the form:

$$V_i(k+1) = V_i(k) + \gamma_{1i}\left(P_i - X_i(k)\right) + \gamma_{2i}\left(G - X_i(k)\right)$$

Where,

Basic Notations of PSO

Parameter	**Summary**
I	Particle Index
K	Discrete time index
V	Velocity of the ith particle
X	Position of ith particle
P	Best position found by ith particle
G	Best position found by swarm
$\gamma_{1,2}$	Random numbers on the interval [0,1] applied to ith particle.

❖ **Algorithm for PSO**

Start with the Initialization of Population
While! (Ngen || Sc)
For p=1 :Np
If fitness Xp> fitness pbestp
Update pbestp = Xp
For
If fitness Xk>gbest
Update gbest = Xk
Next K
For each dimension d
$v_{pd}^{new} = w * v_{pd}^{old} + c_1 * rand_1 * \left(pbest_{pd} - x_{pd}^{old}\right) + c_2 * rand_2 * \left(gbest_d - x_{pd}^{old}\right)$ (3)
v_{pc}
$v_{pd} = \max\left(\min\left(V_{max}, v_{pd}\right.\right.$
(4)
Next d
Next p
Next generation till stop

The particles are first encoding into a bit string S=F1F2….Fn, n=1,2…m and the bit {1} represents for the selected feature from the dataset and the bit string {0} is the non-selected feature from the dataset. Let us suppose in the dataset the available feature set is 10 then set {F1F2F3…..F10} is then analyzed using PSO. Hence on the basis of which pbest is chosen. Now for the final feature selection each of the particles is then updated according to operation.

$$v_{pd}^{new} = w * v_{pd}^{old} + c_1 * rand_1 * \left(pbest_{pd} - x_{pd}^{old}\right) + c$$
$$* \left(gbest_d - x_{pd}^{old}\right) \quad (5$$

$$S\left(v_{pd}^{new}\right) = \frac{1}{1 + e^{-v_{pd}^{new}}} \quad (6)$$

$$\left(rand < S\left(v_{pd}^{new}\right)\right)$$

$$x_{pd}^{new} = 1 \ else \ x_{pd}^{new} = 0$$

Various Notations used in Pseudo Code

Parameter	**Summary**
Ngen	Number of generations or iterations
Sc	Stopping Criteria
Np	Number of particles
Xp	Current position of pheromone
Pbestp	Pheromone with best fitness
x_{pd}^{old}	Previous fitness value
Xk	Current particle position
Gbest	Best fitness value
K	Current particle number
v_{pd}^{new}	Updated particle velocity
v_{pd}^{old}	Current particle velocity
rand1	Random number 1
rand2	Random number 2
c1	Acceleration factor 1
c2	Acceleration factor 2
V_{max}	Maximum Velocity

Therefore, in a PSO method, all particles are initiated randomly and evaluated to compute fitness together with finding the personal best (best value of each particle) and global best (best value of particle in the entire swarm). After that a loop starts to find an optimum solution. In the loop, first the particles' velocity is updated by the personal and global bests, and then each particle's position is updated by the current velocity. The loop is ended with a stopping criterion predetermined in advance [22].

1. For each particle initialize particle
 2. Repeat for each particle
      a). Calculate fitness value
      b). If the fitness value is better than
best fitness value (Pbest) in
  history, set current value as the new Pbest.
3. Choose the particle with the best fitness value of all the particles as the Gbest.
4. For each particle
      a).Update particle velocity according to equation (3.1 a)
      b).Update particle position according to equation (3.1 b)
5. until stopping criteria.

Pseudo code for Particle Swarm Optimization

RANDOM WALKER SEGMENTATION

It is a technique of segmentation on the basis of selecting foreground and background as seed pixels by moving randomly to other pixels moving from background till any foreground pixel is obtained and the region is extracted as segmented region from the image.

IV. Result Analysis

Parameters	Existing Work	Proposed Work
Retrieval Rate	92.68	95.67
Precision	91.65	95.38
Recall	87.26	92.64
F-Measure	89.40114024	93.9900351
Median	97.46	99.75

Table 1. Comparative Analysis on Various Parameters

	Image1	image2	Image3	Image4	Image5
Image1	99.99	0	0	0	0
image2	0	97.42	0	0	0
Image3	0	0	99.83	0	0
Image4	0	0	0	72.52	0
Image5	0	0	0	0	97.55

Table 2. Confusion Matrix of Existing Work

	Image1	image2	Image3	Image4	Image5
Image1	100	0	0	0	0
image2	0	98.65	0	0	0
Image3	0	0	99.94	0	0
Image4	0	0	0	76.38	0
Image5	0	0	0	0	98.75

Table 3. Confusion Matrix of Proposed Work

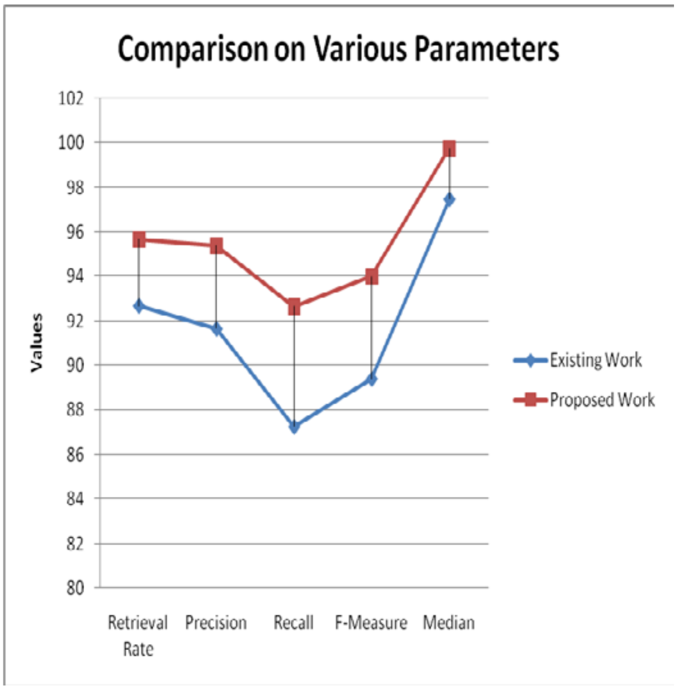

Figure 2. Comparison on Various Parameters

V. Conclusion

The Existing Methodology implemented for the Recognition and Classification of Fooled Images using Deep Neural Network provides efficient classification of fooled images on MNIST Dataset while the technique provides some issues such as increased error rate and accuracy of ~92%. Hence an efficient technique is implemented using Random Walker Segmentation and Optimization of SVM (Support Vector Machine) using PSO (Particle Swarm Optimization) is implemented which provides more accuracy of ~96.

References

[1] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In Computer Vision and Pattern Recognition (CVPR), IEEE Conference on. IEEE, 2015.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (NIPS), pages 1097–1105, 2012.

[3] Thomas Dean, Mark A Ruzon, Mark Segal, Jonathon Shlens, Sudheendra Vijayanarasimhan, and Jay Yagnik. Fast, accurate detection of 100,000 object classes on a single machine. In Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, pages 1814–1821. IEEE, 2013

[4] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 35(7):1757–1772, 2013.

[5] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. DeCAF: A deep convolutional activation

feature for generic visual recognition. In arXiv: 1310.1531, 2013.

[6] Zeiler, M., Taylor, G., and Fergus, R. Adaptive deconvolutional networks for mid and high level feature learning. In ICCV, 2011

[7] A. Nguyen, J. Yosinski, and J. Clune, “Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," 2015.

[8] S. Gu and L. Rigazio, “Towards deep neural network architectures robust to adversarial examples," CoRR, vol. abs/1412.5068, 2014.

[9] Mahendran, A. and Vedaldi, A. Understanding Deep Image Representations by Inverting Them. ArXiv e-prints, November 2014.

[10] Bruna, J., And Mallat, S. Invariant scattering convolution networks. arXiv preprint arXiv:1203.1513. 2012.

[11] Jarrett, K., Kavukcuoglu, K., Ranzato, M., And Lecun, Y 2009. What is the best multi-stage architecture for object recognition? In Computer Vision, IEEE 12th International Conference on, IEEE, 2009, 2146–2153.

[12] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems 25 (F. Pereira, C. Burges, L. Bottou, and K. Weinberger, eds.), pp. 1097-1105, Curran Associates, Inc., 2012.