RI: Small: Deep Learning in Forensic Impression Evidence Comparison

1 INTRODUCTION

This research will address a need of the forensic sciences using recent developments in machine learning. The forensic sciences have been under great stress in recent years with a decline in public and judicial confidence in the abilities of scientists in the courtroom [1, 2]. A case in point is the analysis of impression evidence, e.g., footwear prints, latent prints, handwriting, tire treads, etc. A human expert, i.e., a forensic scientist, performs side-by-side comparison of the evidence with a known, e.g., a questioned signature with known signatures, a crime scene shoe-print with suspect footwear, a latent fingerprint found in a crime scene with a database of fingerprints, etc, and expresses a conclusion as one of three discrete statements: individualization, no opinion and exclusion. In several court cases the final judgment of guilty/innocent has been subsequently found to be in error, e.g., by a later assessment using DNA analysis.

The need to characterize uncertainty of the conclusion in the form of a likelihood ratio (LR) is now recognized as being essential to reporting forensic comparison results [3, 4, 5]. LR is defined as the ratio of the joint probability of evidence and known under two alternative hypotheses(same/different). The LR can be subsequently used with appropriate priors to determine the posterior probability. In the case of handwriting, each of the evidence and known are represented by a set of features by a Forensic Document Examiner (FDE)– the features being chosen for distinguishing between writers [6]. Features for comparing handwritten *th* are shown in Fig. 2.

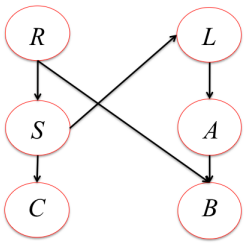
The result of comparing the known feature vector ϕ_k and the evidence feature vector ϕ_e is expressed as an LR. The numerator of the LR, which corresponds to the same source hypothesis h^0, is the joint probability of ϕ_e and ϕ_k when both arise from the same individual. The denominator, which corresponds to the different source hypothesis h^1, is the joint probability of the evidence and

(a) First Writer (b) Second Writer

Figure 1: Handwriting of two individuals (three samples). One set of features extracted per writer.

R = Height Relationship of t to h	L = Shape of Loop of h	A = Shape of Arch of h	C = Height of Cross on t staff	B = Baseline of h	S = Shape of t
r^0= t shorter than h	l^0 = retraced	a^0 = rounded arch	c^0 = upper half of staff	b^0 = slanting upward	s^0 = tented
r^1 = t even with h	l^1 = curved right side and straight left side	a^1 = pointed	c^1 = lower half of staff	b^1 = slanting downward	s^1 = single stroke
r^2 = t taller than h	l^2 = curved left side and straight right side	a^2 =no set pattern	c^2 = above staff	b^2 = baseline even	s^2 = looped
r^3 = no set pattern	l^3 = both sides curved		c^3 = no fixed pattern	b^3 = no set pattern	s^3 = closed
	l^4 = no fixed pattern				s^4 = mixture of shapes

(a) $\phi = [Val(R), Val(L), Val(A), Val(C), Val(B), Val(S)]$.



(b) Bayesian network

Figure 2: Expert features for *th*: (a) values, and (b) a Bayesian network BN_{th} for its probability distribution. The full distribution needs $4,799$ joint probabilities while BN_{th} needs only 100 conditional probabilities [7].

known when they arise from different individuals. Thus we can write

$$LR(\phi_k, \phi_e) = \frac{p[(\phi_k, \phi_e)|h^0]}{p[(\phi_k, \phi_e)|h^1]}. \tag{1}$$

where the numerator and denominator are two distributions, the first conditioned on being from the *same* source and the second from *different* sources. The LR can be used in a Bayesian formulation to provide posterior odds– which is computed as the product of prior odds and LR , i.e.,

$$O_{posterior}(k:e) = LR(\phi_k, \phi_e) \times O_{prior}(k:e) \tag{2}$$

The prior odds can be obtained from the population of suspects, e.g., if there are n equally likely suspects then $p(h^0) = 1/n$, then $p(h^1) = (n-1)/n$ and $O_{prior} \approx 1/n$. Fig. 3 shows the prior and posterior odds for $LR = 10^6$ for different n [5].

Evaluating LR by directly computing (1) is intractable since it requires two joint probabilities in high-dimensional feature space, e.g., for *th* shown in Figure 2 each distribution would have 36 variables requiring 23 million probabilities, and even a Bayesian network that captures conditional independences [8] would be quite complex. One simplification is to use distributions of a kernel

Population Size, n	Posterior Odds With $LR = 1{,}000{,}000$	$P(h^0)$
World (7,000,000,000)	1:7,000	0.0001
USA (300,000,000)	1:300	0.0033
NYC (8,000,000)	1:8	0.1111
Colorado Springs (400,000)	2.5 : 1	0.7143
Walla Walla (30,000)	33:1	0.9706
College Dormitory (200)	5,000:1	0.9998

Figure 3: Different populations of size n and corresponding posterior odds when likelihood ratio is 10^6 (From [5]). The last column shows the posterior odds converted to the probability of h^0.

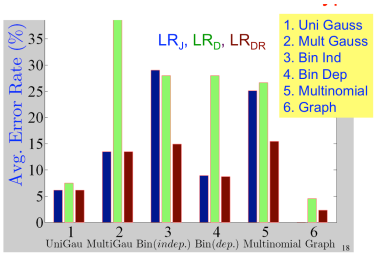


Figure 4: Average error rates of three generative approaches to determine likelihood ratiio: LR(blue), LR_D(green) and LR_{DR}(red). Six data sets were used: univariate Gaussian, multivariate Gaussian, independent binary features, correlated binary features, multinomial distributions and graphs. Overall, LR_D has the highest error and LR_{DR} has the lowest error. The LR method was approximated with independence assumptions, and hence is not the best [11].

function $d(\phi_k, \phi_e)$ leading to a definition LR_D as follows:

$$LR_D(\phi_k, \phi_e) = \frac{p[d(\phi_k, \phi_e)|h^0]}{p[d(\phi_k, \phi_e)|h^1]}. \tag{3}$$

This method is used in some handwriting and speaker verification systems [9]. While it straightforward to evaluate (3), it is a severe approximation of (1) since it effectively projects a multidimensional distribution into a single dimension. A compromise is an approximation that factorizes the LR into similarity and rarity terms, LR_{DR} [10, 11] defined as:

$$LR_{DR}(\phi_k, \phi_e) = p[d(\phi_k, \phi_e)|h^0] \times \frac{1}{p[\frac{1}{2}(\phi_k + \phi_e)]}. \tag{4}$$

The factorization in (4) is intuitively appealing since a human expert gives more importance to unusual features, i.e., low probability (rare) features are given higher weight [12]. This method of LR computation is interestingly analogous to TF-IDF used in information retrieval. However the computational complexity is similar to (1).

Thus we have three methods of LR computation: (i) direct evaluation using (1), (ii) kernel

evaluation using (3), and (iii) factorization using (4). A comparison of the performance of the three methods is shown in Fig. 4 [11]. Each of the three methods have two serious practical limitations, both of which we propose to address in this research.

1. *Intractability:* All three are *generative* probabilistic models which require a full representation of the necessary probability distributions [13]. It is prohibitively expensive to construct for (1), e.g., even for the six variable discrete distribution shown in Fig. 2, we would need 12 variables to represent both ϕ_k and ϕ_e which in turn would need 23 million probabilities for each of the two hypotheses. In general, if we have n features taking k values each, we would need $2k^{2n}$ probabilities, which is exponential in complexity. Even a Bayesian network of $2n$ variables would require a huge numbers of samples.

2. *Human Engineering:* All three rely on hand-crafted features ϕ designed by human experts such as in Fig. 2(a). It is impractical to define different sets of features for comparisons of different handwritten words, footwear types, etc. Features hand-crafted for each application is subject to bias– which is of significant concern to the judicial system.

In this research we propose to overcome intractability by using a *discriminative* approach. Instead of separately learning the two probability distributions in (1), we would train them jointly. A discriminative model would express LR as:

$$LR(\phi_k, \phi_e) = f(\phi_k, \phi_e, \mathrm{w}) \tag{5}$$

where w is a set of parameters. A neural network would determine the parameters as those that maximize the likelihood of a set of training samples, or equivalently minimize the cross-entropy error [1]. The function f for a single hidden layer neural network is given in (6) and (7).

Eliminating human engineered features is possible with *deep learning*, a machine learning approach that has shown spectacular results in many tasks such as object recognition, speech recog-

[1] Likelihood and cross-entropy error are defined as follows. Denoting $\phi_=(\phi_k, \phi_e)$, and given data set $(\phi_n, t_n), n = 1, .., N$ of labeled known-evidence pairs, the likelihood is $p(\mathrm{t}|\mathrm{w}) = \prod_{n=1}^{N} y_n^{t_n}(1-y_n)^{1-t_n}$, where $\mathrm{t} = [t_1, .., t_N]^t$ and for a no-hidden layer neural network (logistic regression) $y_n = \sigma(\mathrm{w}^t\phi_n)$. Cross-entropy error is $E(\mathrm{w}) = -\ln p(\mathrm{t}|\mathrm{w}) = -\sum_{n=1}^{N}[t_n \ln y_n + (1-t_n)\ln(1-y_n)]$. Since there is no closed-form solution for minimizing $E(\mathrm{w})$, gradient descent of the form $\mathrm{w}^{\tau+1} = \mathrm{w}^{\tau} - \eta\nabla E$ is used. The partial derivatives ∇E are determined by error backpropagation, whose complexity is $O(W^2)$, where W is the total number of adaptive parameters.

nition, computer vision, natural language processing, etc. The principal advantage of deep learning is that the necessary representation is automatically learnt in a supervised manner. However the architecture for forensic comparison needs to be fundamentally different. Object recognition typically works with a single image as input and object classes are widely different, e.g., cat versus dog, different people, handwritten digits, etc. Whereas forensic comparison takes a pair of inputs as input with both having many major features in common, e.g., *th* written by two people.

An important extension of the model described is to compute the cumulative LR over several comparisons, e.,g., comparison of multiple handwritten words/notes or several footwear impressions in a crime scene. An example is shown in Fig. 5 where phrases, rather than words, are compared. While methods of fusing multiple classifiers [14] are relevant, explicit methods of combining LRs may be more useful [15]. Recurrent neural networks are also a possible solution [16].

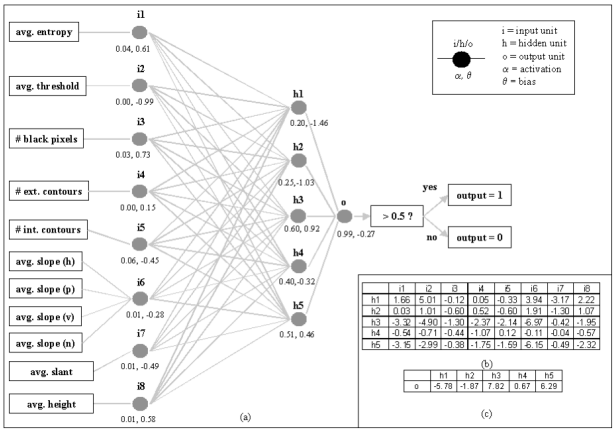
(a) Writer 1 (b) Writer 2

Figure 5: Comparison of phrase of words ($K = 8$) written by two individuals: need to combine eight LRs.

2 OTHER PREVIOUS WORK IN FORENSIC COMPARISON

2.1 Generative Models

With the relatively recent realization that forensic opinion needs to be accompanied with a characterization of uncertainty, several researchers have proposed LR methods. However the methods proposed are all based on either (i) modeling probability distributions of features or (ii) modeling distributions of similarity measures for computing LR_D [3, 5]. A generative method of computing LR_D using (3) with eleven feature differences together with a naiive Bayes model, is used in the CEDAR-FOX software system [9]. The software further maps the LR_D to a discrete opinion scale based on their distribution [18]. The nine-point FDE scale has the opinions: *identified-as-same, highly-probable-same, probably-did, indications-did, no-opinion, indications-did-not, probably-did-not, highly-probable-did-not, identified as elimination* [19].



	i1	i2	i3	i4	i5	i6	i7	i8
h1	1.66	5.01	-0.12	0.05	-0.33	3.94	-3.17	2.22
h2	0.03	1.01	-0.60	0.52	-0.60	1.91	-1.30	1.07
h3	-3.32	-4.90	-1.30	-2.37	-2.14	-6.97	-0.42	-1.95
h4	-0.54	-0.71	-0.44	-1.07	0.12	-0.11	-0.04	-0.57
h5	-3.15	-2.99	-0.38	-1.75	-1.59	-5.15	-0.49	-2.32

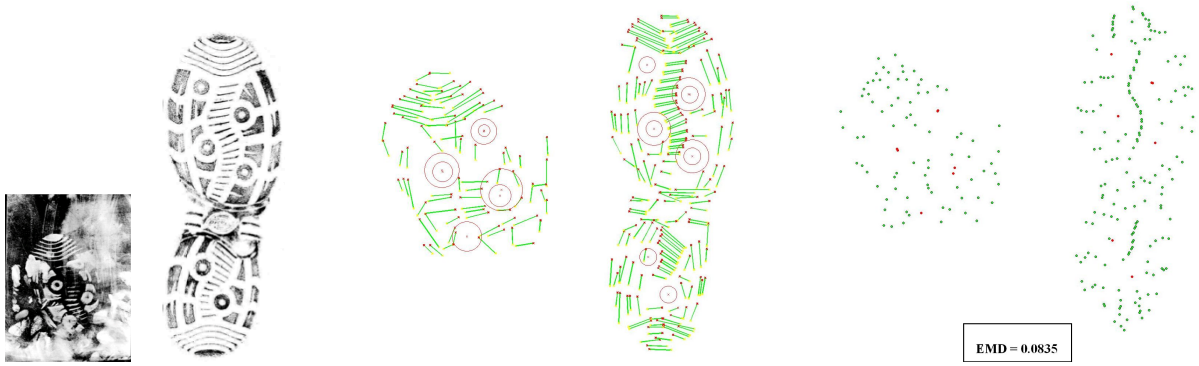
	h1	h2	h3	h4	h5
o	-5.78	-1.87	7.82	0.67	6.29

Figure 6: A neural network for handwriting comparison using feature dissimilarity. Input consists of differences of eleven features. There are five hidden units and a single output unit [17].

2.2 Discriminative Models

Although neural network training using stochastic gradient descent (SGD) has been around for several decades [13], the forensic sciences have not benefited. As an experimental system, a single hidden layer network was used to compare handwriting samples in [17] (Fig. 6)– to determine same/different writer. Input consisted of eleven differences (distances) corresponding to eleven features. This architecture is a discriminative method for computing LR_D, see (3), rather than LR.

The proposed research will move away from computing LR_D (used in both [17] and [9]) and compute the full LR based on a complete representation of ϕ_k and ϕ_e. This will take us into a much higher level of complexity than hitherto attempted in any forensics sub-discipline. The effort is encouraged by the success of deep learning [16], where the system learns representations with multiple levels of abstraction. Deep learning has dramatically improved the state of the art in speech recognition, visual object recognition [20], object detection, drug discovery and genomics. They dispense with careful engineering and domain expertise needed to transform raw data into a feature vector. In particular convolutional networks, where earlier layers alternate between convolutional and pooling layers, which are trained jointly using backward error propagation and SGD, have demonstrated invariance to pose, lighting, background, and surrounding objects. Another approach is to determine high-level features using large-scale unsupervised learning using Restricted Boltzmann Machines (RBM) in a generative way [21].



(a) Evidence (b) Known (c) Features (d) Nodes in graphs

Figure 7: Footwear impression evidence analysis: (a) crime scene print, (b) its known source, (c) detected features (centers of circles) in evidence and known, and (d) graph representations of evidence and known (edges not shown). A modified earthmover distance (EMD) between graphs is used to determine LR_D (from [22, 23]).

2.3 Footwear impression comparison

This problem is a difficult one that has very little automation: at best a human identifies features interactively, which are then used to query a database of footwear outsoles. Our previous effort in developing an algorithm for comparing footwear impressions is shown in Figure 7. In this method, a distance (or similarity) between evidence and known is computed. The distance is based on using computer vision techniques to extract features (centers of circles and ellipses). A relational graph is constructed from these points. Finally a distance between graphs is computed. Thus at best we can determine LR_D rather than LR by using this method. A comprehensive survey and comparison of automated methods for footwear impression comparison can be found in our report [22, 23].

3 PROPOSED RESEARCH AND CAPACITY BUILDING

The proposed work will investigate different deep neural network architectures for forensic comparison, performing experiments with data sets from two forensic impression evidence domains. We begin by noting that (1) can be rewritten as

$$LR(\phi_k, \phi_e) = \frac{p[(\phi_k, \phi_e)|h^0]}{p[(\phi_k, \phi_e)|h^1]} = \frac{p[h^0|\phi_k, \phi_e]p(h^1)}{p[h^1|\phi_k, \phi_e]p(h^0)} = \frac{p(h^0|\phi_e, \phi_k)}{1 - p(h^0|\phi_e, \phi_k)} \tag{6}$$

where we have assumed equal priors, $p(h^0) = p(h^1)$. LR can be directly determined using a neural network with a single hidden layer as follows:

$$p(h^0|\phi) = \sigma_1 \left(\sum_{j=1}^{M} w_j^{(2)} \sigma_2 \left(\sum_{i=1}^{D} w_{ji}^{(1)} \phi_i + b_1 \right) + b_2 \right) \tag{7}$$

where $\phi = (\phi_e, \phi_k)$ has D components; $w_j^{(2)}, w_{ji}^{(1)}, b_1, b_2$ are parameters determined from supervised training data; σ_1 is an activation function (either sigmoid or probit); σ_2 is another activation, possibly ReLU; and M is model complexity (no. of hidden units). A toy example of both models (generative and discriminative) is shown in Fig. 8, where the computed LRs are quite close. Our research will explore architectures with multiple layers, different levels of model complexity.

The task differs from classic object recognition, e.g., handwritten digit recognition, in that we have two inputs rather than one. Interlacing the two inputs ϕ_e, ϕ_k is a significant issue, since the goal is to learn minor differences between objects that are otherwise quite similar. Forensic comparison thus focuses on individuality while ignoring average over the population, i.e, we are interested in the tails of the distribution (of shape). Conversely, object recognition concerns how similar the object is to the mean/mode of the distribution.

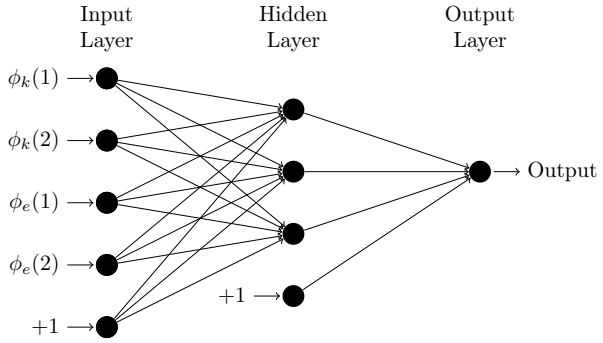
3.1 Deep Learning Architectures

We propose to evaluate several architectures for the forensic comparison task. We will begin with human extracted features for handwriting (as seen in Fig. 2) and automatically extracted features for footwear impressions (as seen in Fig. 7) and then proceed to raw images as input. As we have seen earlier, even when we have human extracted features, the generative approach is infeasible due to exponential complexity. Thus a discriminative solution based on such features itself would be of great benefit to the forensic community. If we eventually succeed with raw images, we would circumvent the need for human and machine-effort in feature extraction.

For these experiments we will explore both conventional neural network and convolutional neural network (CNN) architectures. One design consists of separate networks for inputs ϕ_k and ϕ_e with shared weights, with the results combined at later stages, as shown in Fig. 10. Another design would interlace the two inputs so that minor feature differences can be learnt– while such as design

ϕ_e, ϕ_k	$p\left(\phi_e, \phi_k \mid h^0\right)$	$p\left(\phi_e, \phi_k \mid h^1\right)$
00,00	0.1957	0.0296
00,01	0.0217	0.0740
00,10	0.0326	0.0592
00,11	0.0022	0.1405
01,00	0.0217	0.0592
01,01	0.1739	0.0148
01,10	0.0043	0.0740
01,11	0.0217	0.0740
10,00	0.0283	0.0888
10,01	0.0152	0.1036
10,10	0.1935	0.0296
10,11	0.0435	0.0592
11,00	0.0022	0.0015
11,01	0.0217	0.0888
11,10	0.0217	0.0888
11,11	0.2000	0.0148

(a) Parameters of generative model.



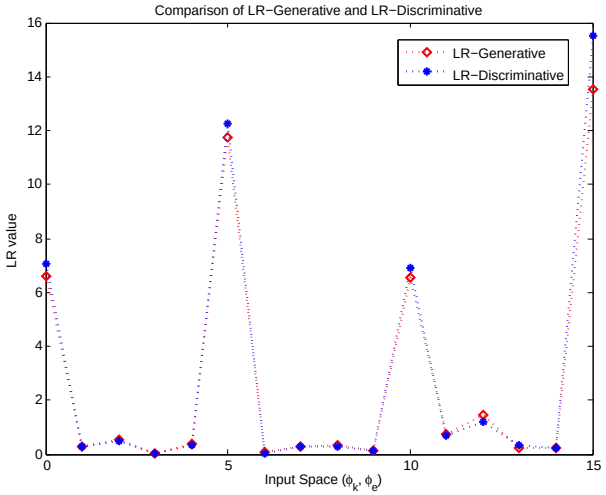
(b) A discriminative model: neural network for computing $p(h^0|\phi_k.\phi_e)$. It was trained using learning-rate = 0.01, mini-batch-size = 100, and no. of iterations = 300. The number of generated samples was 20,000.

w11	-5.5169	2.7104	5.8524	-0.7585
w12	-5.4884	-4.807	4.6924	6.5
w13	0.6106	-5.8483	-3.2345	5.8516
b1	-3.9789	0.5988	-1.6443	
w2	5.9490	-4.3345	5.8192	
b2	0.2064			

(c) Parameters of neural network (weights and biases)

Input	Hidden Node 1	Hidden Node 2	Hidden Node 3
0 0 0 0	0.0184	0.6454	0.1619
0 0 0 1	0.0087	0.9992	0.9853
0 0 1 0	0.8669	0.995	0.0075
0 0 1 1	0.7531	1	0.7257
0 1 0 0	0.2195	0.0147	0.0006
0 1 0 1	0.1164	0.9082	0.1623
0 1 1 0	0.9899	0.6187	0
0 1 1 1	0.9787	0.9991	0.0076
1 0 0 0	0.0001	0.0075	0.2624
1 0 0 1	0	0.8335	0.992
1 0 1 0	0.0255	0.4509	0.0138
1 0 1 1	0.0121	0.9982	0.8297
1 1 0 0	0.0011	0.0001	0.001
1 1 0 1	0.0005	0.0393	0.263
1 1 1 0	0.2823	0.0067	0
1 1 1 1	0.1556	0.817	0.0139

(d) Values of hidden layer activations



(e) Likelihood ratios using the generative model (red) and the discriminative model (blue). The two models show similar values, with peaks corresponding to $\phi_k = \phi_e$.

Figure 8: Two models for determining $LR(\phi_k.\phi_e)$: (a) generative model (16 probabilities) when ϕ_k and ϕ_e are represented by two bits each, (b) discriminative model: neural network with 4-3-1 nodes with output $p(h^0|\phi) = \sigma\left[W_2\sigma\left(W_1\phi + b_1\right) + b_2\right]$, (c) weights and biases of the neural network, (d) activation values, and (e) $LR(\phi_k, \phi_e)$ for the sixteen possible inputs using both models; peak values occur when $\phi_k = \phi_e$ (reflecting exact match), and extremely low values occur when $\phi_k = \overline{\phi_e}$ (exclusion).

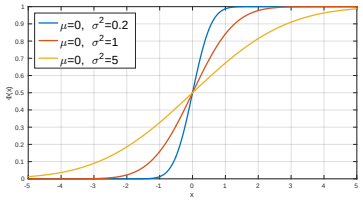


Figure 9: Probit non-linear activation function. It is the cdf of a Gaussian, whose standard deviation is determined from training data

comes closer to our earlier design (Fig. 6), it does not lose information through a distance/kernel function.

First, we will try and see how handwriting comparison performs with a single-hidden-layer densely connected neural network. Then we add a convolution layer together with a pooling layer on top of the input layer and see how performance changes. If performance improves, we can try adding another convolution layer and a pooling layer. Each time we add a new convolution-pooling layer the number of feature maps increases by a certain factor. The final output of all convolution-pooling layers will go into a final densely connected layer. At first, the number of nodes in the densely connected layer is small, so that the model does not overfit the training data in the last layer. Then we gradually increase the number of nodes in the final densely connected layer until performance stops increasing.

Nonlinear activation functions suitable for LR computation will be explored. The ReLU[2] has become popular for earlier stages. A probit for the final output may be useful to avoid extremely large/small LRs. A probit function provides a gradual transition between the two output states (Fig. 9). Since probit is a cdf of a Gaussian, its variance can be determined from training data.

3.2 Learning Algorithms

The principal method to update weights in deep learning is SGD with the key equation $\mathrm{w}^{\tau+1} = \mathrm{w}^{\tau} - \eta \nabla E$, where ∇E, partial derivatives with respect to the weights, is determined in mini-batches using error back propagation and η is the learning rate. Local minima are not a serious problem for deep learning since several minima provide similar results [16]. While SGD is efficient, further speed-up is possible with line search and conjugate descent. Contrastive divergence is useful if we

[2]Rectifier Linear Unit has become most popular for deep learning.

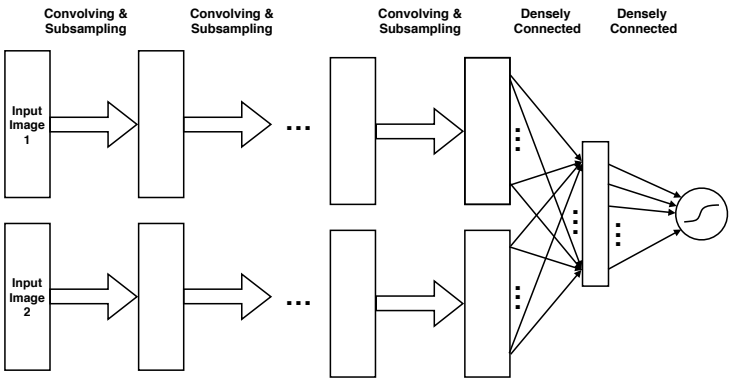


Figure 10: A proposed convolutional neural network for LR computation.

Figure 11: Data augmentation using transformations: translation, rotation, scaling and squeezing.

incorporate into the network a generative model, such as a Restricted Boltzmann Machine (RBM), to determine features.

Training will require several experimental settings, e.g., no. of convolutional layers, no. of nodes in each layer, no. of feature maps, size of filters, size of pooling size, kind of regularization, learning rate, etc. As an example, for learning rate we first choose a small portion of the training data and pick up an initial try and iterate SGD for a few loops. If the loss function is observably decreasing, then we can double the learning rate in the next try. If the loss does not go down in the first few iterations, then perhaps we picked up too big a learning rate and SGD will not converge. The next try will use half of the original value. This process is continued until we find a learning rate that converges fast.

3.3 Regularization

Our preliminary experiments show that simple models tend to overfit, e.g., although the error on the training data is small, error on testing is hard to enhance. Data augmentation is one method of regularization that other researchers have found to be effective. We introduce some distortion or perturbation to produce new data, an example of which is given in Fig. 11.

Standard methods of regularization used in machine learning, e.g., those based on minimizing

the Lagrangian with L_2 , L_1 can also be used. More recently regularization of large fully-connected neural networks by setting a randomly selected subset of activations in each layer to zero, called dropout [24] , and drop-connect [25] (where a randomly selected subset of weights are set to zero) have been proposed. We will experiment with each of these or a combination of them. They are all controlled by a coefficient, which can be determined by observing model performance on a separate validation set.

3.4 Multiple Evidence Combination

We will explore three evidence combination approaches: The first assumes independent comparisons. For K comparisons, with features $\Phi_k = \{\phi_{k_i}\}_{i=1}^K$ and $\Phi_e = \{\phi_{e_i}\}_{i=1}^K$, we get:

$$p(h^0|\Phi_k, \Phi_e) = \frac{p(h^0) \cdot \prod_{i=1}^K LR(\phi_{k_i}, \phi_{e_i})}{p(h^1) + p(h^0) \cdot \prod_{i=1}^K LR(\phi_{k_i}, \phi_{e_i})} \tag{8}$$

where: $LR(\phi_{k_i}, \phi_{e_i}) = \frac{p(\phi_{k_i}, \phi_{e_i}|h^0)}{p(\phi_{k_i}, \phi_{e_i}|h^1)}$. The posterior probability in (8), can be converted into $LR(\Phi_k, \Phi_e)$ by using (6). Since this result is not always satisfactory, e.g., we can get nearly the same value $LR(\Phi_k, \Phi_e)$ based on many comparisons or only a few comparisons, alternative weighting methods can be used [15]. In the second approach dependencies between comparisons use either generative HMMs or discriminative CRFs [8], where each observation and latent node of the PGM would have two variables. The third method is Recurrent Neural Networks (RNN) which have done well in natural language processing[16].

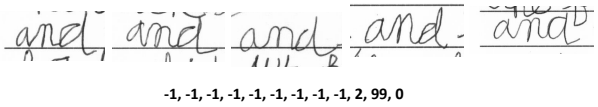
3.5 Data Sets for Training Models

We will use data from two different impression evidence modalities:

1. *Handwriting*: we have several handwriting data sets available for this research:

   (a) Scanned full page writings of 1,500 adults representative of the US population: 3 pages per writer (allows same/different comparisons) [17]. Snippets shown in Fig. 5.

   (b) Paragraphs of handwriting of children in grades 2-4 over a period of three years. Which allows study of growth of handwriting individuality as they grow older [26].



-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 2, 99, 0

(a)

X_1	Initial stroke of "a"	staff right	staff left	staff center	
X_2	Formation of "a" staff	tented	retraced	looped	no staff
X_3	Number of "n" arches	one	two		
X_4	Shape of "n" arches	pointed	rounded	retraced	combination
X_5	Location of "n" mid	above base	below base	at base	
X_6	Formation of "d" staff	tented	retraced	looped	
X_7	Formation of "d" initial	overhand	underhand	straight across	
X_8	Formation of "d" terminal	curved up	straight	curved down	no obvious end stroke
X_9	Symbol	unusual		symbol	
X_{10}	a-n relationship	a taller	a equal	a smaller	
X_{11}	a-d relationship	a taller	a equal	a smaller	
X_{12}	n-d relationship	n taller	n equal	n smaller	

(b)

Figure 12: Handwritten *and* data.: (a) images for a single writer, and (b) features definition table.

(c) Extracted and truthed *th* and *and* images together with FDE-assigned feature values (twelve features for *and* are shown in Fig. 12). They were chosen because *th*, *an* and *nd* are among the the most common bigrams in the English language. There are 888 writing samples from children over three years.

2. *Footwear prints*: Several hundred crime scene and standard footwear prints are available [22]. Some examples are shown in Fig. 7.

Since much of the success of deep learning has come from the availability of huge data sets, we will address it as follows:

1. Augment image data sets by subjecting the given samples to various transformations, e.g., translation, rotation, scaling, deformation.

2. Extract some preliminary features, e.g., for footwear images using methods described in [22], and perform discriminative training above that level.

3.6 GPU implementation

Since CNN training usually involves intensive computation, a GPU for the training task is indispensable. Compared to a CPU, a GPU has a huge amount of computing unit that can perform parallel computation which is very advantageous for matrix operations like convolution. There are various implementations of the GPU library for matrix operations. For our implementation, we

could use Theano [3], which is a Python library that supports GPU computation. One advantage is that it can perform automatic differentiation. The NVIDIA Tesla K40 graphic card is available for $3,000 which we propose to acquire for this research.

4 Project Milestones and Timeline

1. September 1, 2016: Begin studies on neural networks for forensic comparison.
2. December 31, 2016: Complete preparing evidence-known data sets using data augmentation.
3. August 31, 2017: Complete study of neural network methods for (i) human extracted features for handwriting and (ii) machine-extracted features for footwear impressions.
4. August 31, 2018: Complete Deep Learning implementations for raw images of handwriting and footwear impressions.
5. June 31 2019: Refine Deep Learning Methods and Complete Evidence Combination studies.
6. August 31, 2019: Write-up final conclusions and prepare report

5 Capacity Building: Curriculum Development Activities

The PI teaches courses on ML and PGMs. The ML course has seniors and first year undergraduates (135 students for ML and 60 for PGM in 2015). The material is updated each year and the lectures with presentations are made freely available at `http://www.cedar.buffalo.edu/~srihari/CSE574`. Students are assigned several programming projects. Projects related to this research will provide students exposure to forensic problems as well as deep learning.

[3]Theano is a utility package for Python to realize CNN models. We first write code to construct mathematical relationships between symbolic variables, e.g., relations between input, weights, output, loss functions. We can also specify parameters for SGD (epochs, mini-batch size, eta, lambda, etc). Theano will compile C code for such functions to achieve better run-time performance. After that, we can simply use them as a normal Python function and feed in data to compute results.

6 Intellectual Merit

Three new concepts are to be explored: (i) a discriminative model to arrive at a probabilistic statement in forensic comparison, (ii) deep neural networks to work with two inputs rather than one (evidence and known), and (iii) obtaining a single probabilistic opinion using several comparisons.

7 Broader Impacts of the Proposed Work

The forensic sciences are at a crisis mode at present– since many currently used methods, based on human expertise, largely do not satisfy legal requirements of having a scientific basis. The probabilistic method proposed in this research has the potential of providing a dramatic breakthrough, not only to the forensic domains used in this research but also to other forensic domains as well, e.g., latent prints, video evidence, etc.

8 Results from Prior NSF Support

Project ID and Title: IIS-0750876, 2007-09, *Recognition of Handwritten Words in School Essays Using Conditional Random Fields*, $100,000.

8.1 Intellectual Merit

Contextual dependency between words was use in recognition using CRFs. Resulting method was made part of a system to score handwritten essays. Published in: S. N. Srihari, J. Collins, R. K. Srihari, H. Srinivasan, S. Shetty, and J. Brutt-Griffler, "Automatic Scoring of Short Handwritten Essays in Reading Comprehension Tests," Artificial Intelligence, vol. 172(2-3) 2008, pp. 300-324.

8.2 Broader Impacts

First paper to describe how schools can automate scoring of handwritten essays. Success in the task would lead to students receiving performance results earlier in the school year which would enhance their learning.