Neural Networks 97 (2018) 173–182

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Deep neural networks for texture classification—A theoretical analysis☆

Saikat Basu [a,*], Supratik Mukhopadhyay [a], Manohar Karki [a], Robert DiBiano [a], Sangram Ganguly [d], Ramakrishna Nemani [b], Shreekant Gayaka [c]

[a] Louisiana State University, Baton Rouge, LA, USA
[b] NASA Ames Research Center, Moffett Field, CA, USA
[c] Applied Materials, Santa Clara, CA, USA
[d] Bay Area Environmental Research Institute/NASA Ames Research Center, Moffett Field, CA, USA

ARTICLE INFO

Article history:
Received 5 March 2017
Received in revised form 7 August 2017
Accepted 3 October 2017
Available online 23 October 2017

Keywords:
Deep neural network
Texture classification
vc dimension

ABSTRACT

We investigate the use of Deep Neural Networks for the classification of image datasets where texture features are important for generating class-conditional discriminative representations. To this end, we first derive the size of the feature space for some standard textural features extracted from the input dataset and then use the theory of Vapnik–Chervonenkis dimension to show that hand-crafted feature extraction creates low-dimensional representations which help in reducing the overall excess error rate. As a corollary to this analysis, we derive for the first time upper bounds on the VC dimension of Convolutional Neural Network as well as Dropout and Dropconnect networks and the relation between excess error rate of Dropout and Dropconnect networks. The concept of *intrinsic dimension* is used to validate the intuition that texture-based datasets are inherently higher dimensional as compared to handwritten digits or other object recognition datasets and hence more difficult to be shattered by neural networks. We then derive the mean distance from the centroid to the nearest and farthest sampling points in an n-dimensional manifold and show that the *Relative Contrast* of the sample data vanishes as dimensionality of the underlying vector space tends to infinity.

1. Introduction

Texture is a key recipe for various object recognition tasks which involve texture-based imagery data like Brodatz (WWW1, 0000), VisTex (WWW2, 0000), Drexel (Oxholm, Bariya, & Nishino, 2012), KTH (WWW3, 0000), UIUCTex (Lazebnik, Schmid, & Ponce, 2005) as well as forest species datasets (de Paula Filho, Oliveira, & Britto Jr, 2009). Texture characterization has also been shown to be useful in addressing other object categorization problems like the Brazilian Forensic Letter Database (BFL) (Freitas, Oliveira, Sabourin, & Bortolozzi, 2008) which was later converted into a *textural representation* in Hanusiak, Oliveira, Justino, and Sabourin (2012). In Costa, Oliveira, Koerich, and Gouyon (2013), a similar approach was used to find a textural representation of the Latin Music Dataset (Silla Jr., Koerich, & Kaestner, 2008).

Over the last decade, Deep Neural Networks have gained popularity due to their ability to learn data representations in both supervised and unsupervised settings and generalize to unseen data samples using hierarchical representations. A notable contribution in *Deep Learning* is a *Deep Belief Network* (DBN) formed by stacking *Restricted Boltzmann Machines* (Hinton, Osindero, & Teh, 2006). Another closely related approach, which has gained much traction over the last decade, is the Convolutional Neural Network (CNN) (Lecun, Bottou, Bengio, & Haffner, 1998). CNNs have been shown to outperform DBN in classical object recognition tasks like MNIST (WWW4, 0000) and CIFAR (Krizhevsky, 2009). Despite these advances in the field of Deep Learning, there has been limited success in learning textural features using Deep Neural Networks. Does this mean that there is some inherent limitation in existing Neural Network architectures and learning algorithms?

In this paper, following Basu et al., 2016, we try to answer this question by investigating the use of Deep Neural Networks for the classification of texture datasets. First, we derive the size of the feature space for some standard textural features extracted from the input dataset. We then use the theory of Vapnik–Chervonenkis (VC) dimension to show that hand-crafted feature extraction creates low-dimensional representations, which help in reducing the overall excess error rate. As a corollary to this analysis we derive for the first time upper bounds on the VC dimension of Convolutional

☆ This is an extended version of the paper published in International Joint Conference on Neural Networks, IJCNN 2016 (Basu et al., 2016).

* Corresponding author.
E-mail address: sbasu8@lsu.edu (S. Basu).

https://doi.org/10.1016/j.neunet.2017.10.001
0893-6080/

Neural Network as well as Dropout and Dropconnect networks and the relation between excess error rate of Dropout and Dropconnect networks. The concept of *intrinsic dimension* is used to validate the intuition that texture-based datasets lie on an inherently higher dimensional manifold as compared to handwritten digits or other object recognition datasets and hence more difficult to be classified/shattered by neural networks. To highlight issues associated with the Curse of Dimensionality of texture datasets, we provide theoretical results on the mean distance from the centroid to the nearest and farthest sampling points in n-dimensional manifolds and show that the *Relative Contrast* of the sample data vanishes as dimensionality of the underlying vector space tends to infinity. Our theoretical results and empirical analysis show that in order to classify texture datasets using Deep Neural Networks, we need to either integrate them with handcrafted features or devise novel neural architectures that can learn features from the input dataset that resemble these handcrafted texture features.

2. VC dimension of deep neural networks and classification accuracy

VC dimension was first proposed in Vapnik and Chervonenkis (1971) and was later applied to Neural Networks in Bartlett and Maass (2003). It was noted in Bianchini and Scarselli (2014) that the VC dimension proposed for Neural Networks is also applicable to Deep Neural Networks. It was shown in Bartlett and Maass (2003) that for neural nets with sigmoidal activation function, the VC-dimension is loosely upper-bounded by $O(w^4)$ where w is the number of free parameters in the network. Given a classification model M, the VC-dimension of M is the maximum number of samples that can be shattered by M.

We estimate the size of the sample space composed of the various features extracted from the textural Co-occurrence Matrices (Haralick features) following those proposed in Haralick, Shanmugam, and Dinstein (1973). We then use the theory of VC dimension to show that texture feature extraction creates low dimensional representations which help in reducing the overall excess error rate.

2.1. Sample complexity of Haralick features and the fat-shattering dimension[1]

For the sake of simplicity, we consider intensity image with a single channel and Gray-Level Co-occurrence Matrix (GLCM) which can be easily extended to multi-channel images and Color Co-occurrence Matrices (CCM) without loss of generality. For $n \times n$ images with κ color levels, the following results can be derived.

Proposition 2.1. *If $x_1, x_2, \ldots, x_{k^2}$ be the values of the $k \times k$ GLCM matrices, then the number of distinct matrices is given by $\binom{n^2+k^2-1}{k^2-1}$.*

Proof (Sketch). The number of distinct GLCM matrices is the same as the number of non-commutative ways to write n^2 as the sum of k^2 non-negative integers. Assume a line of $n^2 - k^2 + 1$ positions where each position can contain a ball or a divider. If we have n^2 (identical) balls and $k^2 - 1$ dividers we can split the balls into k^2 groups by choosing positions from the dividers: $\binom{n^2+k^2-1}{k^2-1}$. The size of each group corresponds to one of the non-negative integers in the sum. □

Proposition 2.2. *The number of distinct values for GLCM angular 2nd moment is $n^4 - \left(\left\lfloor \frac{n^2}{k^2} \right\rfloor^2 \times \left(k^2-1\right) + \left(n^2 - \left(k^2-1\right)\left\lfloor \frac{n^2}{k^2} \right\rfloor\right)^2 + 1\right)$*

Proof (Sketch). GLCM angular 2nd moment is given by $\sum_i \sum_j p(i,j)^2$. Since the angular 2nd moment can assume only integral values, the number of distinct possible values is given by the range. Maxima occurs when one pixel has the value n^2 and rest are 0. Minima occurs when n^2 is divided between $k^2 - 1$ points and the rest is at the one remaining point. For n^2 divided into k^2-1 points, Number of distinct values is given by $\left(\left\lfloor \frac{n^2}{k^2} \right\rfloor\right)^2 (k^2-1)$ and for the remaining pixel, we have number of distinct values as $(n^2 - \left\lfloor \frac{n^2}{k^2} \right\rfloor^2 (k^2-1))^2$. Adding these two and subtracting it from n^4, we get the final result as $n^4 - \left(\left\lfloor \frac{n^2}{k^2} \right\rfloor^2 \times \left(k^2-1\right) + \left(n^2 - \left(k^2-1\right)\left\lfloor \frac{n^2}{k^2} \right\rfloor\right)^2 + 1\right)$. □

Proposition 2.3. *The number of distinct values of GLCM correlation is $n^2k^2 - n^2 - \frac{k^2}{2} + \frac{k}{2} + 1$.*

Proof (Sketch). GLCM correlation is given by $\frac{\sum_i \sum_j ijp(i,j) - \mu_x\mu_y}{\sigma_x\sigma_y}$. Minima is n^2 when $p(1,1)$ is n^2 and maxima is k^2n^2 when $p(k,k)$ is n^2. The range would then be $k^2n^2 - n^2 + 1$. For other cases, when, $P(k,k)$ is $n^2 - 1$ and $P(k,k-1)$ is 1, Maxima is k^2n^2 and Minima is $k^2n^2 - k$. Similarly, by induction we can show that in the general case, we exclude $\frac{k(k-1)}{2}$ terms. Therefore, a tight upper bound on the solution would be $k^2n^2 - n^2 + 1 - \frac{k(k-1)}{2}$ distinct GLCM correlation values. □

Proposition 2.4. *The number of distinct values of GLCM sum average is $2n^2k - 2n^2 + 1$.*

Proof (Sketch). GLCM sum average is given by $\sum_{i=2}^{2k} ip_{x+y}(i)$ where $p_{x+y}(t) = \sum_{\substack{i=1 \\ i+j=t}}^{k} \sum_{j=1}^{k} p(i,j)$. Maxima is $2n^2k$ which occurs when p(i,j) attains the maximum value, i.e., n^2 at $i = j = k$. Minima is $2n^2$ which occurs when p(i,j) attains the maximum value i.e., n^2 at $i = j = 1$. Therefore, the number of distinct values of GLCM sum average is $2n^2k - 2n^2 + 1$. □

Proposition 2.5. *The number of distinct values of GLCM contrast is $n^2k^2 + n^2 - 2n^2k + 1$.*

Proof (Sketch). GLCM contrast is given by $\sum_{n=0}^{k-1} n^2 \sum_{\substack{i=1 \\ |i-j|=n}}^{k} \sum_{j=1}^{k} p(i,j)$. The maxima occurs when the entire sum n^2 of the GLCM occurs at one pixel at the top right or at one pixel at the bottom left. These are the two points in the GLCM where $|i-j| = k-1$. So, Maxima is $(k-1)^2n^2 = k^2n^2 + n^2 - 2n^2k$. Similarly, Minima occurs when the entire GLCM sum n^2 is distributed among points only along the diagonal where $|i-j| = 0$ hence resulting in a minima of 0. Therefore, distinct number of GLCM contrast values is $n^2k^2 + n^2 - 2n^2k + 1$. □

From Propositions 2.2 through 2.5, it can be seen that in the general case, the number of distinct Haralick features is given by $O(n^2\kappa^2 + n^4)$. For deep neural networks, the VC dimension is upper bounded by $O(w^4)$ according to Bartlett and Maass (2003). Now, we can pick the number of adjustable parameters w to be such that $n \leq \kappa \leq w$ or $\kappa \leq n \leq w$. In both cases, we have $O(n^2\kappa^2) \leq O(w^4)$ and $O(n^4) \leq O(w^4)$ which gives $O(n^2\kappa^2 + n^4) \leq O(w^4)$. Hence, the number of possible distinct values for the GLCM based feature vectors is much lower than the VC dimension of such a network. So,

[1] For a detailed description of the various GLCM metrics defined in this section and the notations used, we refer the reader to Haralick et al. (1973).

we can effectively argue that the VC-dimension of a Deep Neural Network with w adjustable parameters is such that it can shatter the metrics formed using GLCM — the only prerequisite being that we select a network with the number of adjustable parameters as an upper bound for the input data dimensionality and the number of distinct gray levels in the color channel. On the other hand, in order to shatter the raw image vectors, the effective VC dimension of the network should be at least of the order of $O(\kappa^{n^2})$. So, for the GLCM based features, we need Neural Networks with smaller VC dimension as compared to raw vectors. Also, in the next section, we show that with increase in VC dimension of the network, the excess error rate increases. So, the composite learning model formed by the integration of GLCM based features and Deep Neural Networks have lower excess error rate as compared to Deep Neural Networks combined with raw image pixels.

3. Input data dimensionality and bounds on the test error

In this section, we derive the relation between input data dimensionality and upper bound Γ on the excess error rate of the Deep Neural Network. This validates the fact that the lower dimensional representations of the Haralick feature space help in minimizing the test error rate. As a corollary to this analysis we derive for the first time upper bounds on the VC dimension of Convolutional Neural Network as well as Dropout and Dropconnect networks and show that the upper bound Γ on the excess error rate of the Dropout networks is lower than that of DropConnect.

Lemma 3.1. *With increase in the dimensionality of the input data, the dimensionality of the optimal model increases.*

Proof (Sketch). As shown in Makhoul, Schwartz, and El-Jaroudi (1989), for input data dimensionality d and model dimensionality p, the number of cells formed by p planes in d space is given by

$$C(p,d) = \sum_{i=0}^{min(p,d)} \binom{p}{i} = \begin{cases} 2^p & , p \le d \\ \sum_{i=0}^{d} \binom{p}{i} & , p > d. \end{cases} \tag{1}$$

Now, the number of cells per dimension gives the number of divisions of the model space along each dimension and can be approximated as $C(p,d)^{1/d}$. This in turn is equal to the number of class labels c. Therefore, for a given classification problem with c class labels, we have $C(p,d)^{1/d} = c$ and hence, we have

$$c = C(p,d)^{1/d} = \begin{cases} 2^{p/d} & , p \le d \\ (\sum_{i=0}^{d} \binom{p}{i})^{1/d} & , p > d. \end{cases} \tag{2}$$

From Eq. (2), it follows that with increase in data dimensionality d, the model dimensionality p should increase, given a fixed classification problem with c class labels. □

Lemma 3.2. *With increase in the dimensionality of the model, its VC dimension increases.*

Proof (Sketch). This statement follows from the VC dimension bounds of both a Deep Neural Network and a Deep Convolutional Neural Network (CNN). The VC dimension of a Deep Neural Network is upper bounded by $O(w^4)$ and the VC dimension of a Convolutional Neural Network is upper bounded by $O\left(\frac{m^4k^4s^{2l-2}}{l^2}\right)$. The result for the Deep Neural Network follows from Bartlett and Maass (2003), where it is noted that the VC dimension of Deep Neural Networks with sigmoidal activation functions is given by $O(t^2d^2)$ which reduces to $O(w^4)$. The result of the VC bound for the CNN along with the proof is detailed in Theorem 3.3. □

Theorem 3.3. *The VC dimension of a Convolutional Neural Network is upper bounded by $O\left(\frac{m^4k^4s^{2l-2}}{l^2}\right)$ where m is the total number of maps, k is the kernel size, s is the subsampling factor and l is the number of layers.*

Proof (Sketch). From Theorems 5 and 8 in Bartlett and Maass (2003), it can be seen that for the parameterized class $F = \{x \longmapsto f(\theta, x) : \theta \in \mathbb{R}^d\}$ with the arithmetic operations $+, -, \times, /$ and the exponential operation $\alpha \longmapsto e^{\alpha}$, jumps based on $>, \ge, <, \le, =$ and $\ne$ evaluations on real numbers and output 0/1, VCDim(F) $= O(t^2d^2)$. Here, t is the number of operations and d is the dimensionality of the adjustable parameter space. Now, for the CNN, input size is n, kernel size is k, sampling factor is s and we assume convolution kernel step size as 1 for simplicity. So, we have $\frac{\frac{n-k}{s}-k}{s}-k \ldots$ up to l layers which in turn is equal to 1 for a binary classification problem.[2] Now, in the simplest case, we have a CNN with one convolutional layer followed by one subsampling layer (c–s). Hence, $\frac{n-k}{s} = 1 \Longrightarrow n = s + k$. For a CNN with the configuration (c–s–c–s), we have

$$\frac{\frac{n-k}{s}-k}{s} = 1 \Longrightarrow n = k + s(s+k) = s^2 + ks + k. \tag{3}$$

Continuing this pattern, we have in the general case,

$$\begin{aligned} &\frac{\frac{\frac{n-k}{s}-k}{s}-k}{s} \ldots \text{ up to } l \text{ layers} = 1 \\ &\Longrightarrow n = s^l + ks^{l-1} + ks^{l-2} + \cdots + ks + k. \end{aligned} \tag{4}$$

Now, let $m_1, m_2, \ldots, m_l$ be the number of maps in the various layers of a CNN and $t = t_1 + t_2 + \cdots + t_l$ be the total number of operations. Now, for layer 1, number of operations $t_1 = m_1(n-k)$, for layer 2, number of operations $t_2 = m_2(\frac{n-k}{s} - k)$, and so on. Therefore, Total number of operations

$$\begin{aligned} t &= m_1(n-k) + \cdots + m_l(\frac{\frac{n-k}{s}-k}{s} \ldots \text{to } l \text{ layers}) \\ &= m_1(ks + ks^2 + \cdots + ks^{l-1} + s^l) + m_2(ks + ks^2 + \cdots \\ &\quad + ks^{l-2} + s^{l-1}) + \cdots + m_l s. \end{aligned} \tag{5}$$

Also, dimensionality of parameter space is given by $d = m_1k + m_2k + \cdots + m_lk$. Now, for simplifying, if we assume that the number of maps in the layers $m_1 = m_2 = \cdots = m_l = \frac{m}{l}$, then, we have

$$\begin{aligned} t &= \frac{m}{l}(n-k) + \frac{m}{l}(\frac{n-k}{s} - k) + \cdots + \frac{m}{l}(\frac{\frac{n-k}{s}-k}{s} - k \\ &\quad \ldots \text{ up to } l \text{ layers}) = \frac{mks^2(s^{l-1}-1)}{l(s-1)^2} + \frac{ms(s^l-1)}{l(s-1)} \\ &= O(\frac{mks^{l-1}}{l}) \end{aligned} \tag{6}$$

$$\text{Also, } d = O(mk). \tag{7}$$

From Eqs. (6) and (7), we have $\text{VCdim}_{CNN} = O\left(\frac{m^4k^4s^{2l-2}}{l^2}\right)$. □

Theorem 3.4. *Upper bound on excess error rate $\mathcal{E}$ increases with increase in VC dimension given fixed number of training samples N.*

Proof (Sketch). According to the theory of VC dimension (Vapnik, 1996), we have Excess error rate

$$\mathcal{E} \le \sqrt{\frac{h(\log(2N/h)+1) - \log(\eta/4)}{N}} \tag{8}$$

[2] Note that for simplifying the algebra, we consider only the convolutional and subsampling layers of a CNN. This analysis can be extended to hybrid architectures with other types of layers (e.g., fully connected) by adjusting t and d.

where h is the VC dimension of the model, N is the number of training samples and $0 \leq \eta \leq 1$. From Eq. (8), the result follows. □

Theorem 3.5. *For a given Dropout network with probability of dropout p and number of adjustable parameters in the network being w, the VC dimension of the network is upper bounded by $O\Big((1-p)^8 w^4\Big)$.*

Proof (Sketch). For a neural network with number of neurons $n = n_1 + n_2 + n_3 + \cdots + n_l$, the number of adjustable parameters w is given by

$$w = n_1 n_2 + n_2 n_3 + n_3 n_4 + \cdots + n_{l-1} n_l. \tag{9}$$

For a given dropout fraction p, each neuron in the network can be dropped by a probability of p. So the effective number of neurons in the Dropout network

$$\widetilde{n} = (1-p)(n_1 + n_2 + \cdots + n_l). \tag{10}$$

Now, we can split the effective number of neurons in each layer as $\widetilde{n_1} = (1-p)n_1, \ldots, \widetilde{n_l} = (1-p)n_l$.

$$\begin{aligned} \text{Therefore, } \widetilde{w} &= \widetilde{n_1}\widetilde{n_2} + \widetilde{n_2}\widetilde{n_3} + \cdots + \widetilde{n_{l-1}}\widetilde{n_l} \\ &= (1-p)^2(n_1 n_2 + \cdots + n_{l-1} n_l) = (1-p)^2 w. \end{aligned} \tag{11}$$

Now, given that $\text{VCDim}_{Dropout} = O\Big(\widetilde{w}^4\Big)$ we have $\widetilde{w}^4 = ((1-p)^2)^4 w^4 = O\Big((1-p)^8 w^4\Big)$. So,

$$\text{VCDim}_{Dropout} = O\Big((1-p)^8 w^4\Big). \quad \square \tag{12}$$

Theorem 3.6. *For a given Dropconnect network with probability of drop p and number of adjustable parameters in the network being w, the VC dimension of the network is upper bounded by $O\Big((1-p)^4 w^4\Big)$.*

Proof (Sketch). Since in a Dropconnect network, each weight can be dropped by a probability of p, so, effective number of adjustable parameters in the Dropconnect network is given by $\widetilde{w} = (1-p)w$. Now, given that $\widetilde{w}^4 = (1-p)^4 w^4 = O\Big((1-p)^4 w^4\Big)$, we have

$$\text{VCDim}_{Dropconnect} = O\Big((1-p)^4 w^4\Big). \quad \square \tag{13}$$

Theorem 3.7. *For a given drop probability p, the number of adjustable parameters in the network being w, the excess error rate being $\mathcal{E}$ and the upper bounds on the error rates of the Dropout and Dropconnect networks being $\Gamma_{Dropout}$ and $\Gamma_{Dropconnect}$, respectively, we have $\Gamma_{Dropout} \leq \Gamma_{Dropconnect}$.*

Proof (Sketch). From Eqs. (8) and (12), we have

$$\text{Excess error rate, } \mathcal{E} \leq \sqrt{\frac{h(\log(2N/h)+1) - \log(\eta/4)}{N}}. \tag{14}$$

Therefore, upper bound on the error rate $\Gamma_{Dropout}$

$$= \sqrt{\frac{1}{N}(1-p)^8 w^4 \Big[\log\Big(\frac{2N}{(1-p)^8 w^4} + 1\Big)\Big] - \log(\eta/4)}. \tag{15}$$

Similarly, from Eqs. (8) and (13), we have for the Dropconnect network, $\Gamma_{Dropconnect}$

$$= \sqrt{\frac{1}{N}(1-p)^4 w^4 \Big[\log\Big(\frac{2N}{(1-p)^4 w^4} + 1\Big)\Big] - \log(\eta/4)}. \tag{16}$$

Table 1
Intrinsic dimension estimation using MLE on the MNIST, CIFAR-10 and DET datasets.

Dataset	MNIST	CIFAR10	DET
Intrinsic Dim.	9.96	15.9	17.01

Table 2
Intrinsic dimension estimation using MLE on the 6 texture datasets.

Dataset	Brodatz	VisTex	KTH
Intrinsic dimension (raw vectors)	34.87	44.81	43.69
Intrinsic dimension (texture features)	4.03	3.84	3.73
Dataset	KTH2	Drexel	UIUCTex
Intrinsic dimension (raw vectors)	54.19	30.26	33.64
Intrinsic dimension (texture features)	3.93	4.24	4.57

For a given w, N and probability of drop p with $0 \leq p < 1$, it can be easily shown that the upper bounds on the excess error rates of the Dropout and Dropconnect networks are related as

$$\Gamma_{Dropout} \leq \Gamma_{Dropconnect}. \quad \square \tag{17}$$

This is substantiated by the experimental results in Section 6.

4. What is the difference between object recognition datasets and texture-based datasets in terms of dimensionality?

We argue that object recognition datasets lie on a much lower dimensional manifold than texture datasets. Hence, even if Deep Neural Networks can effectively shatter the raw feature space of object recognition datasets, the dimensionality of texture datasets is such that without explicit texture-feature extraction, these networks cannot shatter them. In order to estimate the dimensionality of the datasets, we use the concept of *intrinsic dimension* (Levina & Bickel, 2004).

4.1. Intrinsic dimension estimation using the maximum likelihood algorithm

The *intrinsic dimension* of a dataset represents the minimum number of variables that are required to represent the data. We use the Maximum Likelihood algorithm proposed in Levina and Bickel (2004) to estimate the Intrinsic dimension of various datasets. The results for the various datasets and the Haralick features extracted are listed in Tables 1 and 2. The DET dataset (Russakovsky et al., 2015) is a subset of the Imagenet dataset. As shown in Levina and Bickel (2004), the Intrinsic Dimension of a dataset increases with increase in sample size considered given a particular dataset. However, they did not provide any results highlighting the relationship between intrinsic dimension and sample across multiple datasets. In order to avoid ambiguities, we select multiple rounds of the same sample size for each dataset and average the final results to get the intrinsic dimension of the entire dataset.

From Tables 1 and 2, we can see that the intrinsic dimensionality of the texture datasets (Brodatz, VisTex, KTH, KTH2, Drexel and UIUCTex) is much higher than that of object recognition datasets (MNIST, CIFAR-10 and DET). So, without explicit texture-feature extraction, a deep neural network cannot shatter the texture datasets because of their intrinsically high dimensionality. However, as seen in Table 2, the features extracted from the texture datasets have a much lower intrinsic dimensionality and lie on a much lower dimensional manifold than the raw vectors and hence can be shattered/classified even by networks with relatively smaller architectures. Once, we have validated the fact that texture-based datasets lie on a higher dimensional manifold as compared to handwritten digit or object recognition datasets, we highlight issues associated with the high dimensionality of texture datasets.

5. Curse of dimensionality in texture datasets

Curse of Dimensionality refers to the phenomenon where classification power of the model decreases with increase in dimensionality of the input feature space. In the following sections, we derive some theoretical results on *Curse of Dimensionality* for high-dimensional texture data.

5.1. Sampling data in higher dimensional manifolds

The mean distance from the centroid to the nearest sampling point is a useful metric for quantifying the hardness of classification (Hastie, Tibshirani, & Friedman, 2001). To compute this mean distance, we first state a result on computing the expected value of a non-negative random variable and then use it to compute the mean distance from the centroid to the nearest sample point. The median distance was computed in Hastie et al. (2001). However, to get a more accurate estimate of the distance metrics, we compute the mean in this paper.

Lemma 5.1. *If a random variable y can take on only non-negative values, then the mean or expected value of y is given by $\int_0^\infty [1 - F_x(t)]dt$.*

Proof (Sketch). Since $1-F_X(x) = P(X \geq x) = \int_x^\infty f_X(t)dt$, it follows that $\int_0^\infty (1 - F_X(x)dx) = \int_0^\infty P(X \geq x)dx = \int_0^\infty \int_x^\infty f_X(t)dtdx$. Changing the order of integration, we have $\int_0^\infty (1 - F_X(x)dx) = \int_0^\infty \int_0^t f_X(t)dxdt = \int_0^\infty x[f_X(t)]_0^t dt = \int_0^\infty tf_X(t)dt$. Now, taking the substitution $t = x$ and $dt = dx$, the expected value

$$E(X) = \int_0^\infty (1 - F_X(x)dx) = \int_0^\infty (1 - y^p)^n dy. \quad \square \tag{18}$$

Lemma 5.2. *Consider n samples distributed uniformly in a p-dimensional hypersphere of radius 1 and center at (0,0). If at the origin, we consider a nearest neighbor estimate, then the mean distance from the origin to the nearest sampling point is $\prod_{\xi=1}^{n}(1 + \frac{1}{p\xi})^{-1}$.*

Proof (Sketch). For a ball of radius r in R^p the volume is given by $\omega_p r^p$, where ω_p is denoted as $\frac{\pi^{p/2}}{(p/2)!}$ So, the probability of a point sampled uniformly from the unit ball lying within a distance x of the origin is the ratio of the volume of that ball to the volume of the unit ball. The common factors of ω_p cancel, so we get the Cumulative Distribution Function (CDF) and Probability Density Function (PDF) as $F(x) = x^p$, and$f(x) = px^{p-1}, 0 \leq x \leq 1$. From Hogg, McKean, and Craig (2005), for n points with CDF F and PDF f, we have the following general formula for the ξ^{th} order statistic

$$g_k(y_\xi) = \frac{n!}{(\xi - 1)!(n - \xi)!}[F(y_\xi)]^{\xi-1}[1 - F(y_\xi)]^{n-\xi} f(y_\xi). \tag{19}$$

So, we have the minimum by setting $\xi = 1$ as

$$g(y) = n(1 - F(y))^{(n-1)} f(y) = n(1 - y^p)^{n-1} py^{p-1}. \tag{20}$$

This yields the CDF, $G(y) = 1 - (1 - y^p)^n$. The random variable y can take on only non-negative values. So, by Lemma 5.1 the mean or expected value is $E[X] = \int_0^\infty [1 - G_x(t)]dt$. Now, by substituting x^p by z, we have $E(X) = \frac{1}{p}\int_0^1 z^{\frac{1}{p}-1}(1 - z)^n dz$. (Note the change of limits since z lies in [0,1].)

This can be reduced using the Euler Gamma function as $E(X) = \frac{1}{p}.\frac{\Gamma(\frac{1}{p})\Gamma(n+1)}{\Gamma(n+1+\frac{1}{p})}$. Now, by using the identity $\Gamma(z + 1) = z\Gamma(z)$ recursively, we get Mean Distance,

$$D(p, N) = E(X) = \prod_{\xi=1}^{n}(1 + \frac{1}{p\xi})^{-1}. \quad \square \tag{21}$$

Table 3
Mean distance from origin to nearest sampling point for various object recognition and texture datasets.

Dataset	MNIST	CIFAR-10	DET	Brodatz	VisTex
$D(p, N)$	0.32	0.49	0.54	0.74	0.79
Dataset	KTH	KTH2	Drexel	UIUCTex	
$D(p, N)$	0.78	0.79	0.63	0.69	

Table 3 shows the mean distance from the origin to the nearest sampling point for various datasets. From the table and according to Hastie et al. (2001), most data points for the texture datasets are nearer to the feature space boundary than to any other data point. This makes prediction particularly difficult for these datasets because we cannot interpolate between data points and we need to extrapolate. Next, we propose a result on the expected distance from the origin to the farthest data point and then use it to derive the relation of the *Relative Contrast* of the data points to the underlying dimensionality of the vector space as highlighted in Section 5.2.

Lemma 5.3. *Consider n samples distributed uniformly in a p-dimensional hypersphere of radius 1 and center at (0,0). If at the origin, we consider a nearest neighbor estimate, then mean distance from origin to the farthest data point is $1 - \frac{np}{(np+p-1)(np+p)}$.*

Proof (Sketch). Using Eq. (19), and setting $\xi = n$ for the maxima, we have

$$g(y) = n[F(y)]^{n-1} f(y) = ny^{pn-1} py^{p-1} = npy^{pn+p-2}. \tag{22}$$

Therefore, the corresponding CDF is given by $G(y) = np\frac{y^{pn+p-1}}{pn+p-1}$. By Lemma 5.1, the mean or expected value is $E[X] = \int_0^\infty [1 - np\frac{y^{pn+p-1}}{pn+p-1}]dy$

$$= 1 - \frac{np}{(np + p - 1)(np + p)}. \quad \square \tag{23}$$

5.2. Relative contrast in high dimensions

In Beyer, Goldstein, Ramakrishnan, and Shaft (1999), it was shown that as dimensionality increases, the distance to the nearest neighbor approaches that of the farthest neighbor, i.e., contrast between points vanishes, while, in Aggarwal, Hinneburg, and Keim (2001) it was shown that *Relative Contrast* varies as $\sqrt{p}$ for $n = 2$ sample points with dimensionality p. In this paper, we generalize this to the case of n data points and also provide an exact estimate of the *Relative Contrast* instead of providing approximation bounds as Aggarwal et al. (2001). We then show that as dimensionality $p \to \infty$, it yields the same result as Aggarwal et al. (2001) and Beyer et al. (1999). Also, we eliminate the arbitrary constant C used in Aggarwal et al. (2001) which can vary significantly with change in parameters resulting in a fluctuating bound. It should be noted that we assume the L_2 norm distance metric and the Euclidean space for deriving our algebra.

Theorem 5.4. *If $RC_{n,p}$ be the Relative Contrast of n uniformly distributed sample points with p being the dimensionality of the underlying vector space, then, $RC_{n,p} = \frac{1-\frac{np}{(np+p-1)(np+p)}-\prod_{\xi=1}^{n}(1+\frac{1}{p\xi})^{-1}}{\prod_{\xi=1}^{n}(1+\frac{1}{p\xi})^{-1}}$ and $RC_{n,p}$ approaches 0 as p approaches ∞.*

Proof (Sketch). From Lemma 5.2, we can see that the mean distance from the origin to the nearest sampling point is given by the expression $\prod_{\xi=1}^{n}(1 + \frac{1}{p\xi})^{-1}$. And from Lemma 5.3, the mean

Table 4
Test error of a Convolutional Neural Network trained using supervised backpropagation on the various texture datasets.

Texture datasets	Brodatz	Drexel	KTH
CNN 3C test error (%)	28.96	35.27	34.93
CNN 5C test error (%)	78.3	55.31	69.5
AlexNet test error (Krizhevsky, Sutskever, & Hinton, 2012) (%)	91.7	76.97	89.87
BVLC CaffeNet test error (Jia et al., 2014) (%)	89.82	75.92	89.7
Texture datasets	KTH2	UIUCTex	VisTex
CNN 3C test error (%)	40.29	49.75	26.68
CNN 5C test error (%)	85.68	91.18	68.67
AlexNet test error (Krizhevsky et al., 2012) (%)	91.8	93.72	83.7
BVLC CaffeNet test error (Jia et al., 2014) (%)	90.8	91.6	82.9

Table 5
Test error of a Convolutional Deep Belief Network on the various texture datasets.

Texture datasets	Brodatz	Drexel	KTH
CDBN 2 layer test error (%)	62.01	53.52	59.15
CDBN 3 layer test error (%)	65.57	59.71	67.06
CDBN 4 layer test error (%)	90.10	78.26	89.05
Texture datasets	KTH2	UIUCTex	VisTex
CDBN 2 layer test error (%)	82.24	70.02	53.85
CDBN 3 layer test error (%)	73.68	69.44	62.95
CDBN 4 layer test error (%)	90.65	91.79	88.01

distance from the origin to the farthest data point is given by the expression $1-\frac{np}{(np+p-1)(np+p)}$. Therefore, $\frac{E[Dmax-Dmin]}{E[Dmin]}$

$$= \frac{1-\frac{np}{(np+p-1)(np+p)}-\prod_{\xi=1}^{n}(1+\frac{1}{p\xi})^{-1}}{\prod_{\xi=1}^{n}(1+\frac{1}{p\xi})^{-1}}. \quad (24)$$

Now, it can be easily shown that

$$\lim_{p\to\infty}\frac{1-\frac{np}{(np+p-1)(np+p)}-\prod_{\xi=1}^{n}(1+\frac{1}{p\xi})^{-1}}{\prod_{\xi=1}^{n}(1+\frac{1}{p\xi})^{-1}}=0. \quad (25)$$

Therefore, it follows that $\frac{E[Dmax-Dmin]}{E[Dmin]}\to 0$ as $p\to\infty$. Therefore, $RC_{n,p}\to 0$ as $p\to\infty$. From Eq. (24), it can be concluded that for the general case of n sample points with a dimensionality of p, the expected value of the *relative contrast* for the sample points varies as $p^{-(n+1)}$. □

Theorem 5.5. *For $n=2$ the general result proposed in Theorem 5.4 approaches the bound of $\frac{C}{\sqrt{p}}\sqrt{\frac{1}{2\xi+1}}$ proposed in Aggarwal et al. (2001) as the dimensionality p of the underlying sample space approaches ∞.*

Proof (Sketch). From Aggarwal et al. (2001), it can be seen that for dimensionality of p and L_k norm,

$$\lim_{p\to\infty}E[\frac{Dmax-Dmin}{Dmin}.\sqrt{p}]=C\sqrt{\frac{1}{2\xi+1}}. \quad (26)$$

Subtracting the rightmost term in Eq. (24) from Eq. (26), we have $RC_{diff}=RC_{Agg}-RC_{Ours}$

$$=\frac{C}{\sqrt{p}}\sqrt{\frac{1}{2\xi+1}}-\frac{1-\frac{np}{(np+p-1)(np+p)}-\prod_{\xi=1}^{n}(1+\frac{1}{p\xi})^{-1}}{\prod_{\xi=1}^{n}(1+\frac{1}{p\xi})^{-1}}. \quad (27)$$

Therefore, for any arbitrary constant C and a given ξ,

$$\lim_{p\to\infty}RC_{diff}=\lim_{p\to\infty}\Big(\frac{C}{\sqrt{p}}\sqrt{\frac{1}{2\xi+1}}-\frac{1-\frac{np}{(np+p-1)(np+p)}-\prod_{\xi=1}^{n}(1+\frac{1}{p\xi})^{-1}}{\prod_{\xi=1}^{n}(1+\frac{1}{p\xi})^{-1}}\Big). \quad (28)$$

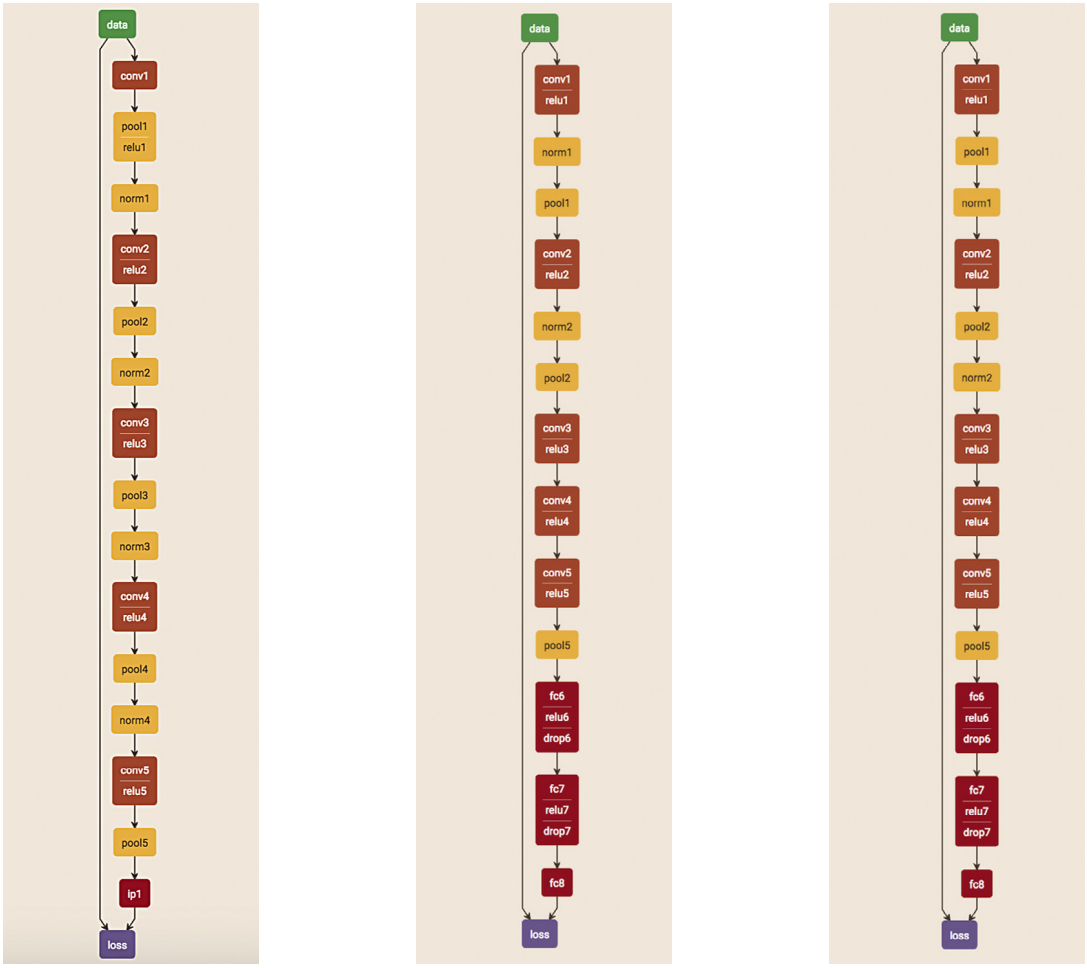
Therefore, by substituting $n=2$ in Eq. (28), it is easy to show that $\lim_{p\to\infty}RC_{diff}=0$. □

Theorem 5.4 validates the result in Beyer et al. (1999) and Theorem 5.5 shows that for the special case $n=2$, our result approaches the bound of Aggarwal et al. (2001) as dimensionality p approaches ∞. So, from Section 4 and Theorem 5.4, we conclude that texture datasets lie on an inherently higher dimensional manifold than object recognition datasets, so their Relative Contrast is lower.

6. Experiments

To validate our theory that error rate for networks with texture features is lower than that of raw vectors, we performed experiments on 6 benchmark texture classification datasets — Brodatz, VisTex, Drexel, KTH-TIPS, KTH-TIPS2 and UIUCTex. We extracted 27 features based on the GLCM metrics presented in Section 2. Along with the GLCM features we also use features extracted from the local spectral histogram of images pre-filtered with different filters like Laplacian of Gaussian with different filter sizes and standard deviation σ, and Gabor filters with both the sine and cosine components following the ideas from Liu and Wang (2003). As highlighted in Liu and Wang (2003), these filters provide efficient ways of extracting spatial structures at different orientations and frequencies. The GLCM based features have also been shown to be useful descriptors for satellite image datasets in Basu, Ganguly, Mukhopadhyay, et al. (2015) and Basu, Ganguly, Nemani, et al. (2015). Without loss of generality, we select image size n^3 to be 28 and number of color levels κ as 256. Also, datasets with multiple color channels are converted to grayscale. The Deep Neural Networks are trained by stacking—(1) Restricted Boltzmann Machines (RBM) and (2) Denoising Autoencoders (SDAE). Both the models are then discriminatively fine-tuned with supervised backpropagation. Fig. 2 shows the schematic of our texture feature based network. Figs. 3 and 4 show the final test error of the backpropagation algorithm on the labeled test data using stacked Restricted Boltzmann Machine (RBM) and Stacked Denoising Autoencoder (SDAE) for unsupervised pre-training. Fig. 5 shows the final test error on a deep neural network without unsupervised pre-training. Table 4 shows the final test error on the various texture datasets using Convolutional Neural Networks with different architectures. The first two CNNs we explore have 3 and 5 convolutional layers, respectively, with 32, 32 and 64 feature maps for the first network and 32, 32, 64, 64 and 64 maps for the next. Each uses 5×5 kernels and are accompanied with max-pooling layers with 3×3 kernels. Each pooling layer is followed by a layer with Rectified Linear units and a local response normalization layer with a 3×3 locality. We use a softmax based loss function and a learning rate which is initially set to 0.001 and then decreased as the inverse power of a gamma parameter (0.0001). Other than these networks, we also use two well-known CNN architectures

[3] Note that we extract $n\times n$ sliding window blocks from the various texture datasets for uniformity of analysis.



(a) 5 layer CNN. (b) AlexNet. (c) BVLC Reference Caffenet.

Fig. 1. Network architectures of various Convolutional Neural Networks used in our experiments. (Figures generated using Netscope Neural Network visualizer WWW5, 0000.)

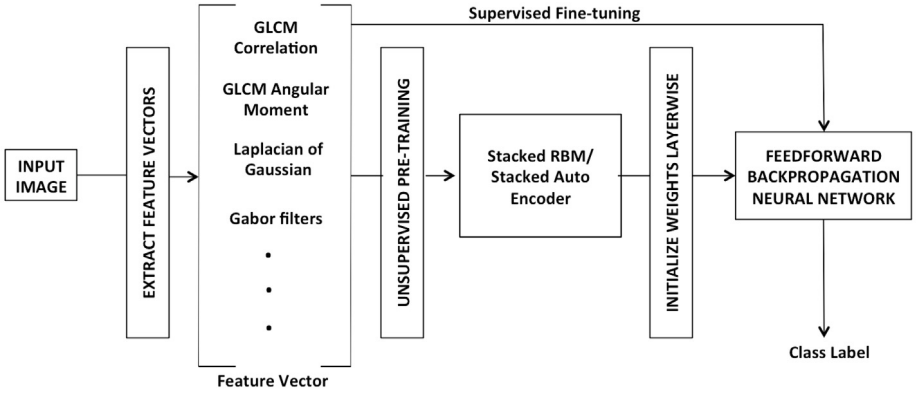
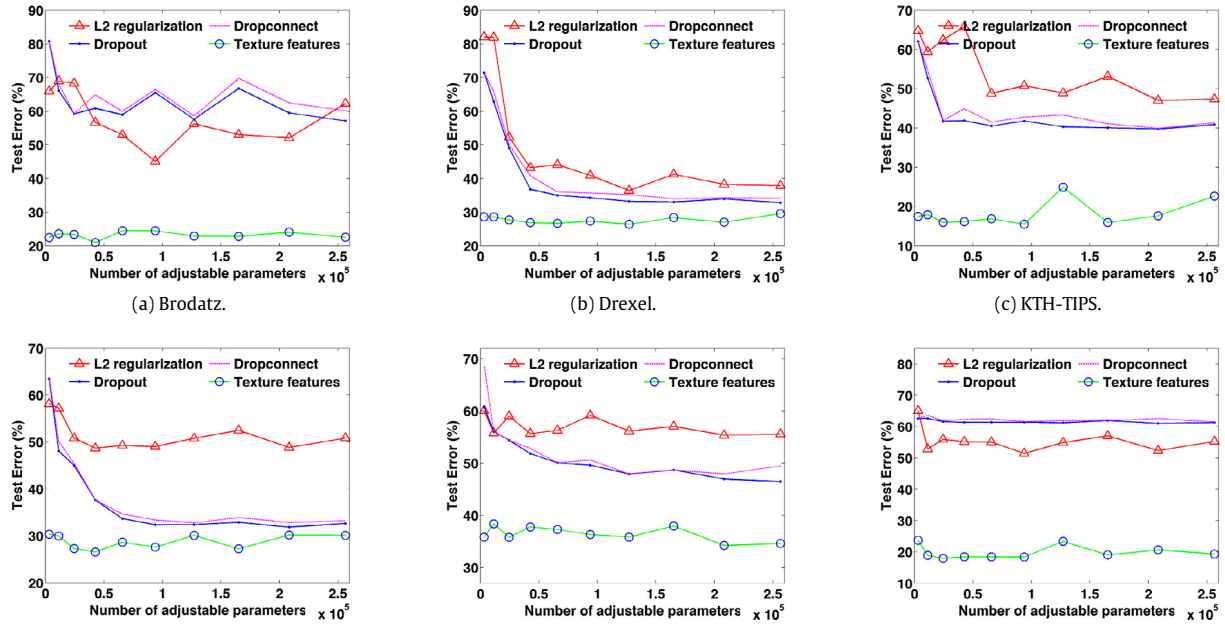


Fig. 2. Schematic of the texture feature based network.

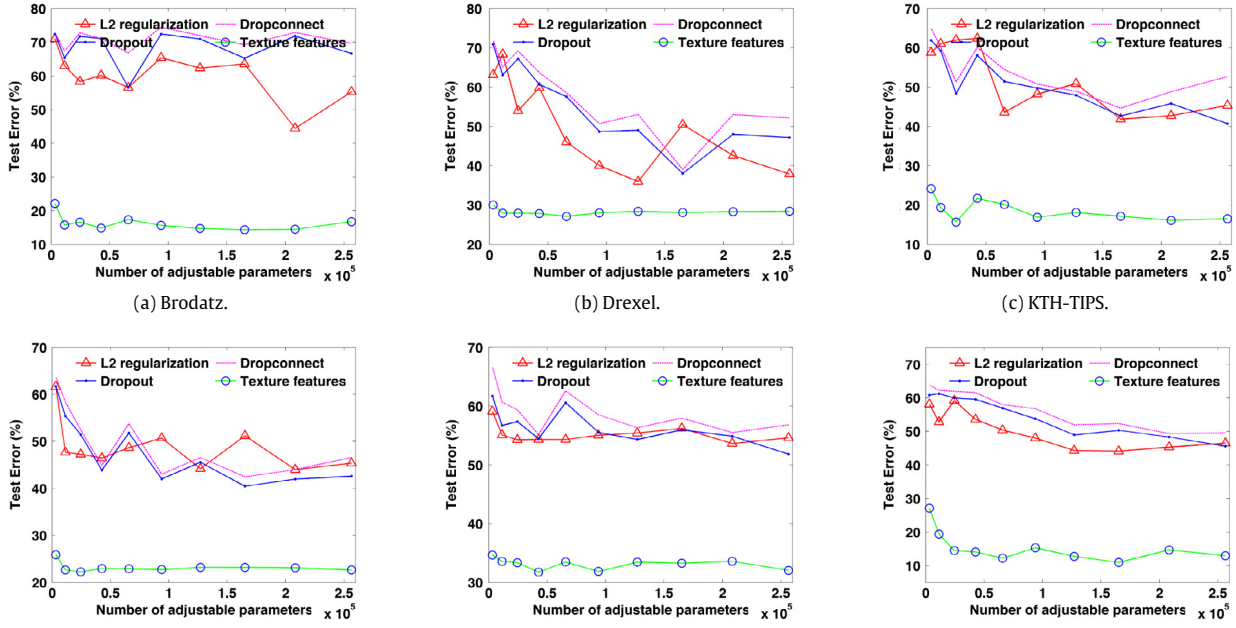
namely AlexNet (Krizhevsky et al., 2012) and BVLC CaffeNet (Jia et al., 2014). The network architecture of the 5 layer CNN, AlexNet and BVLC CaffeNet are shown in Fig. 1. Table 5 shows the test error rates of a convolutional deep belief network (CDBN) (Lee, Grosse, Ranganath, & Ng, 2009) on the various texture datasets. We use three CDBN architectures with 2, 3 and 4 layers. The 2 layer CDBN has 64 and 128 feature maps, the 3 layer CDBN has 64, 64 and 128 feature maps and the 4 layer network has 64 feature maps in the first 3 layers and 128 feature maps in the last layer. We use a filter size of 3×3, stride 1 and pooling stride of 2. The learning rate is set to 0.01 in all the layers before the last layer and the last layer has a learning rate of 0.001. By comparing the results in Figs. 3–5 and Tables 4 and 5, we can see that for all texture datasets, the texture-feature based networks outperform the networks based on raw pixels. Additionally, from Tables 4 and 5, we see that the deeper networks like AlexNet, BVLC Caffenet and the 4 layer CDBN show



(a) Brodatz. (b) Drexel. (c) KTH-TIPS.

(d) KTH-TIPS2. (e) UIUCTex. (f) VisTex.

Fig. 3. Test Error on the 6 texture datasets with the Haralick features and stacked Restricted Boltzmann Machines with L_2 norm regularization, Dropout and Dropconnect obtained by varying the number of adjustable parameters.

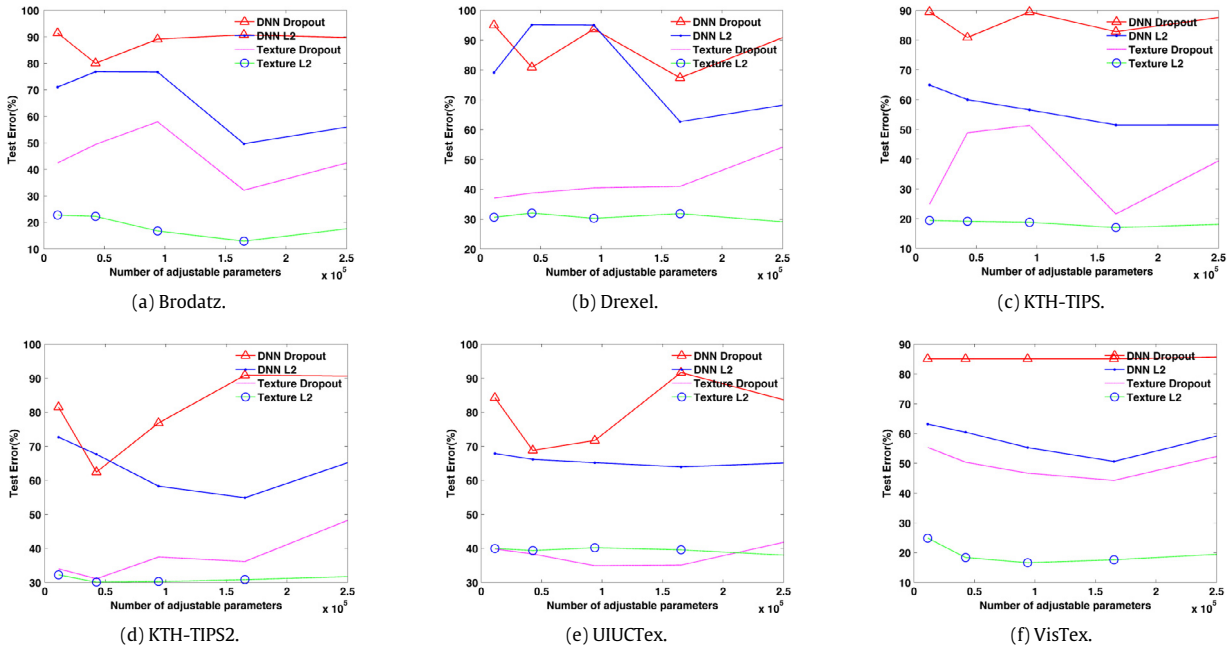


(a) Brodatz. (b) Drexel. (c) KTH-TIPS.

(d) KTH-TIPS2. (e) UIUCTex. (f) VisTex.

Fig. 4. Test Error on the 6 texture datasets with the Haralick features and Stacked Denoising Autoencoders with L_2 norm regularization, Dropout and Dropconnect obtained by varying the number of adjustable parameters.

higher test error rate on all the datasets. This can be attributed to the fact that the bigger networks overfit on these datasets with limited amount of labeled data. Interestingly, it can also be seen from Tables 4 and 5 that for the largest dataset that we consider in our experiments namely Drexel, the bigger CNN networks like the 5 layer CNN, AlexNet and CaffeNet and the bigger 4 layer CDBN network shows the lowest test error rate among all the datasets considered, which substantiates the intuition that if we have significant amount of labeled training data, then it is possible to obtain lower test error rates using only supervised learning on raw image data without the need for hand-crafted texture feature extraction. A closer investigation of Figs. 3–5 shows that for deep neural networks pre-trained by stacking restricted Boltzmann machines and denoising autoencoders as well as those trained only using supervised backpropagation and no unsupervised pre-training, the texture-feature based networks provide lower test error rates than those with raw pixels. So, the experiments substantiate our theoretical claim that extraction of certain hand-crafted texture features create low-dimensional representations that enable Deep Neural Networks to achieve lower test error rate.



(a) Brodatz. (b) Drexel. (c) KTH-TIPS.

(d) KTH-TIPS2. (e) UIUCTex. (f) VisTex.

Fig. 5. Test Error on the 6 texture datasets with the Haralick features and Deep Neural Networks without unsupervised pre-training with L_2 norm regularization and Dropout obtained by varying the number of adjustable parameters.

7. Conclusion

The use of Deep Neural Networks for texture recognition has seen a significant impediment due to a lack of thorough understanding of the limitations of existing Neural architectures. In this paper, we provide theoretical bounds on the use of Deep Neural Networks for texture classification. First, using the theory of VC-dimension we establish the relevance of handcrafted feature extraction. As a corollary to this analysis, we derive for the first time upper bounds on the VC dimension of Convolutional Neural Network as well as Dropout and Dropconnect networks and the relation between excess error rates. Then we use the concept of *Intrinsic Dimension* to show that texture datasets have a higher dimensionality than color/shape based data. Finally, we derive an important result on *Relative Contrast* that generalizes the one proposed in Aggarwal et al. (2001). From the theoretical and empirical analysis, we conclude that for texture data, we need to redesign neural architectures and devise new learning algorithms that can learn features similar to GLCM and other spatial correlation based texture-features from input data.

Acknowledgments

This research was supported by NASA Carbon Monitoring System through Grant #NNH14ZDA001-N-CMS and Cooperative Agreement Number NASA-NNX12AD05A, CFDA Number 43.001, for the project identified as "Ames Research Center Cooperative for Research in Earth Science and Technology (ARC-CREST)". Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect that of NASA or the United States Government. We are thankful to the reviewers for their comments and suggestions which helped in improving the manuscript.

References

Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In *ICDT* (pp. 420–434). Springer.

Bartlett, P. L., & Maass, W. (2003). Vapnik-Chervonenkis dimension of neural nets.

Basu, S., Ganguly, S., Mukhopadhyay, S., DiBiano, R., Karki, M., & Nemani, R. (2015). Deepsat: A learning framework for satellite imagery. In *Proceedings of the 23rd SIGSPATIAL International conference on advances in geographic information systems* (p. 37). ACM.

Basu, S., Ganguly, S., Nemani, R. R., Mukhopadhyay, S., Zhang, G., Milesi, C., et al. (2015). A semiautomated probabilistic framework for tree-cover delineation from 1-m NAIP imagery using a high-performance computing architecture. *IEEE Transactions on Geoscience and Remote Sensing*, *53*(10), 5690–5708.

Basu, S., Karki, M., DiBiano, R., Mukhopadhyay, S., Ganguly, S., & Nemani, R. et al., (2016). A theoretical analysis of deep neural networks for texture classification, In *International joint conference on neural networks*.

Beyer, K. S., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). *ICDT'99. When is "nearest neighbor" meaningful?* (pp. 217–235). London, UK, UK: Springer-Verlag.

Bianchini, M., & Scarselli, F. (2014). On the complexity of neural network classifiers: a comparison between shallow and deep architectures. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(8), 1553–1565.

Costa, Y., Oliveira, L., Koerich, A., & Gouyon, F. (2013). Music genre recognition using gabor filters and lpq texture descriptors. In *Iberoamerican congress on pattern recognition*.

de Paula Filho, P. L., Oliveira, L. S., & Britto Jr., A. S. (2009). A database for forest species recognition, In *Proceedings of the XXII Brazilian symposium on computer graphics and image processing*.

Freitas, C., Oliveira, L. S., Sabourin, R., & Bortolozzi, F. (2008). Brazilian forensic letter database. In *11th International workshop on frontiers on handwriting recognition, Montreal, Canada*.

Hanusiak, R., Oliveira, L., Justino, E., & Sabourin, R. (2012). Writer verification using texture-based features. *International Journal on Document Analysis and Recognition (IJDAR)*, *15*(3), 213–226.

Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural features for image classification. In IEEE Transactions on Systems, Man and Cybernetics*SMC-3*(6), 610–621.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *Springer series in statistics. The elements of statistical learning*. New York, NY, USA: Springer New York Inc.

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*(7), 1527–1554.

Hogg, R. V., McKean, J. W., & Craig, A. T. (2005). *Introduction to mathematical statistics*. Pearson Education.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., et al. (2014). Caffe: convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM International conference on multimedia* (pp. 675–678). ACM.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images, Tech.rep.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 1097–1105.

Lazebnik, S., Schmid, C., & Ponce, J. (2005). A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(8), 1265–1278.

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 2278–2324.

Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th annual International conference on machine learning* (pp. 609–616). ACM.

Levina, E., & Bickel, P. J. (2004). Maximum likelihood estimation of intrinsic dimension. In *NIPS*.

Liu, X., & Wang, D. (2003). Texture classification using spectral histograms. *IEEE Transactions on Image Processing*, *12*(6), 661–670.

Makhoul, J., Schwartz, R., & El-Jaroudi, A. (1989). Classification capabilities of two-layer neural nets, In *1989 International conference on acoustics, speech, and signal processing, 1989. ICASSP-89*, Vol.1, (pp. 635–638). http://dx.doi.org/10.1109/ICASSP.1989.266507, ISSN: 1520-6149.

Oxholm, G., Bariya, P., & Nishino, K. (2012). The scale of geometric texture. In *LNCS*: *Vol. 7572. Computer vision*. ECCV 2012, (pp. 58–71). Springer Berlin Heidelberg.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 1–42. http://dx.doi.org/10.1007/s11263-015-0816-y.

Silla Jr., C. N., Koerich, A. L., & Kaestner, C. A. A. (2008). The latin music database, *Proceedings of the 9th international conference on music information retrieval, Philadelphia, USA*, (pp. 451–456).

Vapnik, V. (1996). Structure of statistical learning theory. *Computational Learning and Probabilistic Reasoning*, *3*.

Vapnik, V. N., & Chervonenkis, A. Y. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, *16*(2), 264–280.

WWW1, 0000. Brodatz, http://sipi.usc.edu/database/database.php?volume=textures.

WWW2, 0000. VisTex,http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html.

WWW3, 0000. KTH, http://www.nada.kth.se/cvap/databases/kth-tips/index.html.

WWW4, 0000. MNIST, http://yann.lecun.com/exdb/mnist/.

WWW5, 0000. Netscope neural network architecture visualizer, http://ethereon.github.io/netscope.