Technical Journal, University of Engineering and Technology (UET) Taxila, Pakistan Vol. 25 No. 4-2020
ISSN:1813-1786 (Print) 2313-7770 (Online)

Extended Relational Autoencoders for Feature Extraction in CBIR

M. K. Bashir[1], Y. Saleem[2], S. Naseer[3]

[1] *Department of Statistics & Computer Science, University of Veterinary & Animal Sciences, Lahore, Pakistan,*
[2] *Department of Computer Science & Engineering, University of Engineering and Technology, Lahore, Pakistan,*
[3] *Department of Computer Science, University of Management and Technology, Lahore, Pakistan.*

[1] mkbashir@uvas.edu.pk

Abstract- As high dimensional data is increasing day by day, importance of fast and precise features extraction is also increasing. Huge work in deep learning leads to autoencoders that are very useful methods for feature extraction as they tend to recreate the original data from same features. In deep learning, a lot depends on loss and objective function. Normally loss functions depend on input data without any relationship between input data. In this study, an extended relational model for autoencoders has been proposed that maps original data on the basis of data with combination to relation between data. This relation is based on ratio between data variance and magnitude. Convolutional autoencoder has been evaluated with proposed relational model in this study. Benchmark datasets of Mnist and Cifar10 have been used for experimental results. Comparison of proposed model has been made with different available loss functions and experimentally it has been proved that proposed relational model achieve lower construction loss with better accuracy and visual results.

Keywords- Surface plasmons, Surface plasmon resonance (SPR), Metamaterials, Localized surface plasmon resonance (LSPR), Plasmonic superabsorber (PSA), Nanocomposites (NCs), Nanoparticles (NPs).

I. INTRODUCTION

Image indexing is very important for image search and retrieval from large databases as well as on internet. Image retrieval from such databases is using tags attached with images. Manual tagging can always be suspicious so we are always in need of such system that can tag images with proper information as per image contents. Content based image retrieval (CBIR) can be used for such retrieval with two main tasks, first taks is generating such features that can represent image contents and secondly converting these features to some code that can be used to retrieve image[1].

Data we have in form of images, videos is multi-dimensional data and converting it into some code is dimensionality reduction. Though many machine learning techniques are available for dimensionality reduction and working very well but number of other issues have also been seen like computational complexity, overfitting, model complications and leads to problem known as curse of dimensionality[2].

Various proposed methods used for dimensionality reduction [3-7] can be divided in two main categories: feature extraction and feature selection. In feature selection, a subset of original features set is used that is always smaller than the original features set. While for feature extraction a new set of features are produced from raw data that are different and smaller from original data. Techniques like Subset Selection [8] and Random Forest [9] are used effectively for filtering out unnecessary features but problem is that all the features are not contributing for final results. So, we can say that we loss some information that does not contribute in final decision and could be important for decision making. As such techniques fails to use all the features so research of dimensionality reduction has moved towards feature extraction to much extent.

Different methods have been proposed to convert high-dimensional data to low-dimensional with minimum information loss that mainly based on projection [10]. Some of these projection methods are LDA (Linear Discriminant Analysis) [11] and PCA (Principal Component Analysis) [12]. Both LDA and PCA are dimensionality reduction techniques that use linear transformation techniques. PCA does not use label information so fall in unsupervised learning technique. Its main task is to find principal components so it uses variance maximization to find the direction of principal components. LDA is known as supervised learning technique as it uses label information. Main goal of LDA is to find the axis information that maximize the distance between multiple classes along with directions. Later on different kernel functions [13-16] have been used to overcome these issues.

For conversion of high-dimensional data into low-dimensional, one needs to maintain uniform relationships among data samples and new features. These relationships are crucial and main focus of dimensionality reduction study. Similarities for visualization for data samples and new features were used as relationship in Multidimensional Scaling (MDS) [17] while ISPMAP [18] extract low-dimensional features by having same distances between data samples. LLE (Locally Linear Embedding) [19] maintain data relationships with involvement of local neighborhood while extracting low-dimensional information. In same way LE (Laplacian Eigenmaps) [20] uses pairwise distance information by minimizing it. Main issue of these techniques is that they use fixed and predefined relationships while this relationship could vary in different parts of image. Another problem faced is that these techniques extract features at once and there is no deeper level information and representations that could be more effective for data retrieval.

Deep neural networks can also be used for dimensionality reduction and autoencoders [21] are the ones that reduce dimensionality by minimizing reconstruction loss. They extract representations and tries to reproduce the same information from these representations using two steps. This is the main reason that autoencoders as well as its extensions like Sparse Autoencoders [22], Denoising Autoencoders [23], Contractive autoencoders [24] and Variational Autoencoders [25] are used to extract meaningful representations from given data. GAE (Generalized Autoencoders) [26] have been used weighted relational function that minimizes distances between reconstructed instances and original ones. GAE has some drawbacks like calculating pair-wise distance, for all the weights, is very challenging and risky as some relationships could be emphasized while some are neglected and is very likely to lose some information. Solution of this problem has been proposed in RAE (Relational Autoencoders) [27]as RAE involves both features and their relationships while minimizing loss. Weak and trivial relationships have been filtered out by adding activation function.

In this study, conventional autoencoders have been used to extract image representations that are further processes to obtain original images. Extensive experimentation has been performed on different benchmark datasets that produced favorable results. In remaining paper, Section II presents review of literature with some hashing and features extraction techniques. Section III presents our methodology in detail with activation and loss functions. Section IV contains experiments details and Section V concludes this study.

II. Literature Review

AE (Autoencoders) [28] were introduced for dimensionality reduction with linear functions and aiming to learn simpler representations. AE is unsupervised learning technique that encoded the representations with minimum loss. In [29] authors investigated and found that one layer autoencoders and PCA are same for dimensionality reduction but computationally AE requires more resources than PCA. But later on nonlinear activation functions were involved that make AE more useful and capable to extract good features [30]. Till this time computation time was biggest problem for deep neural model but in 2006 [31] proposed deep network architecture i.e. Restricted Boltzmann Machine (RBM) and achieved optimization for global gradient. Using similar greedy layerwise method Bengio [32] trained stacked autoencoder for MNIST. Later on different studies [33-36] in this field showed that stacked autoencoders are capable of learning abstract features and meaningful representation for better classification results. Further improvements were made afterwards [37] to increase sparsity of network structure that restrict the weights increment. To impose penalty on large weights regularization in loss function [38] was proposed. Denoising Autoencoders are proposed by Vincent et. al. [39] that used noises to solve this issue. They added Gaussain noise to input that was processed to extract original input from corrupted input.

Representations produced by these autoencoders somehow contain major information from original data that does not make them sensitive to variation in original information. CAE (Contractive Autoencoder) [24] was proposed to overcome this issue and make them sensitive to variation by maintaining mutual relationships between data samples. GAE (Generalized Autencoders) were proposed in [26] that results data relationships not data features. GAE uses distance weights to maintain data relationships that can achieve better results but again it could be biased when assigning pre-defined distance weights. To overcome this problem, a new loss function was proposed in [27] with RAE (Relation Autoencoders) that aims to minimize data representations as well as relationships.

II. Preliminaries

In this section basic autoencoder models for feature extraction will be discussed. These models will be used in next section with hashing to produce image tags.

3.1. Feature Extraction

In this method, original data from high-dimensional space is transformed to low-dimensional space using some linear or nonlinear function. Let if we have given a dataset D with N number of samples and M number of features. Let F_o be the original feature set while a function A transforms F_o to generate new feature set F_n and in any case $|F_n| < |F_o|$.

3.2. Basic Autoencoder

Autoencoder has two main parts: encoder and decoder. Encoder is used to extract image representations or transforming original high-dimensional features to low-dimensional features while decoder works in vice versa. Encoder is a function A that transforms input dataset D to D'(hidden representations). This function *A* can be formulated as

$$D' = f(D) = A(WD + b_D) \quad (1)$$

A function could be linear or nonlinear. If it is an identity function that audoencoder will be extracting features linearly else it will work nonlinearly. *W* represents weight matrix while *b* is bias vector. Decoder function *B* is used to remap extracted representations back to original dataset *D*. Its mathematical presentation will be as

$$D = f(D') = B(W'D' + b_{D'}) \quad (2)$$

We will use the features extracted by encoder step of autoencoder but *D* = *D*' should be verified. Objective function of an autoencoder is given in eq.(3).

$$\emptyset = \min_{\emptyset} L(D, D') = \min_{\emptyset} L(D, g(f(D))) \quad (3)$$

We need to find the parameter W, b_D, $b_{D'}$ that are mainly used to minimize reconstruction loss. Two reconstruction losses are used named as L_1 and L_2. L_1 works for linear reconstruction and generalized from squared error while for nonlinear reconstruction L_2 is used that is derived from cross-entropy.

$$L_1(\emptyset) = \sum_{i=1}^{n} \|x_i - x'_i\|^2 = \sum_{i=1}^{n} \|x_i - g(f(x_i))\|^2 \quad (4)$$

$$L_2(\emptyset) = -\sum_{i=1}^{n} [x_i \, log(y_i) + (1 - x_i) log\,(1 - y_i)] \quad (5)$$

3.3. Relational Autoencoder (RAE)

As discussed earlier that autoencoders generates representation by minimizing reconstruction loss of data only and its drawbacks are covered by RAE that minimizes reconstruction loss of data as well as relationships of the data. Objective function of RAE can be mathematically formed as:

$$\emptyset = (1 - \alpha) \min_{\emptyset} L(D, D') + \alpha \min_{\emptyset} L(R(D), R(D')) \quad (6)$$

where R(D) representing relationship between original data samples while R(D') is relationship between samples of extracted representations. α is scale parameter that helps in controlling reconstruction loss of data and its relationship. There are number of ways to model data relationship but in RAE, it based on similarities of original and extracted data. In this case objective function is given as under:

$$\emptyset = (1 - \alpha) \min_{\emptyset} L(D, D') + \alpha \min_{\emptyset} L(DD^T, (D'D'^T)) \quad (7)$$

A rectifier function [46] was used as activation function to make this algorithm computationally efficient and discarding unnecessary relationships. Rectifier function can be as under:

$$\tau_t(r_{ij}) = \begin{cases} r_{ij}, & if\ r_{ij} \geq t \\ 0 & otherwise \end{cases} \quad (8)$$

t is a threshold to differentiate weak and trivial relationship. So, the final objective function of RAE [27] is

$$\emptyset = (1 - \alpha) \min_{\emptyset} L(D, D') + \alpha \min_{\emptyset} L(\tau_t(DD^T), \tau_t(D'D'^T)) \quad (9)$$

α is a scalar parameter that has been used to control data and relationship losses as shown in eq. DD^T is same as magnitude of D and D^T will be same. In our observation data relationship should require some additional parameters like standard deviation or variance between data. We used coefficient of variation with a little modification to avoid problem of gradient. Equation for coefficient of variation[40] is given in eq. (10).

$$CV = \frac{Standard\ Deviation}{Mean} = \frac{\sigma}{\mu} \quad (10)$$

Standard deviation σ is square root of variance that has been explained in following equations.

$$\sigma^2 = \frac{\sum(D - \mu)^2}{N} \quad (11)$$

$$= \frac{\sum(D^2 - 2\mu D + \mu^2)}{N} \quad (12)$$

$$= \frac{\sum D^2}{N} - \frac{2\mu \sum D}{N} + \frac{N\mu^2}{N} \quad (13)$$

$$= \frac{\sum D^2}{N} - 2\mu^2 + \mu^2 \quad (14)$$

$$= \frac{\sum D^2}{N} - \mu^2 \quad (15)$$

$$= \frac{\sum D^2}{N} - \left(\frac{\sum D}{N}\right)^2 \quad (16)$$

$$\sigma = \sqrt{\frac{\sum D^2}{N} - \left(\frac{\sum D}{N}\right)^2} \quad (17)$$

So, equation of coefficient of variation from eq. (10) will become:

$$CV = \frac{\sqrt{\frac{\sum D^2}{N} - \left(\frac{\sum D}{N}\right)^2}}{mean\ (D)} \quad (18)$$

In eq. (18) if mean is close to zero or zero then there will be problem, even if mean is smaller proposed function will not tend to produce minimum value. As we are going to use it for our loss function, that should be minimized, we used sum of data instead of mean. So, eq. (18) will become

$$\frac{\sqrt{\frac{\sum D^2}{N} - \left(\frac{\sum D}{N}\right)^2}}{sum(D)} \quad (19)$$

Using eq. (19) for data relationship with squared error function, proposed objective function will become as under in eq. (20):

$$\emptyset = (1-\alpha)\min_{\emptyset} L(D, D') + \alpha \min_{\emptyset} L\left(\frac{\frac{\sum D^2}{N} - \left(\frac{\sum D}{N}\right)^2}{sum(D)}, \frac{\frac{\sum D'^2}{N} - \left(\frac{\sum D'}{N}\right)^2}{sum(D')}\right) \quad (20)$$

IV. Extended Relational Autoencoders

As explained earlier different types of Autoencoders are there that can be used for dimensionality reduction and generating original image from same reduced data. In this study we have used extended version of relational convolutional autoencoder for features extraction and classification.

4.1. Extended Relational Convolutional Autoencoder (ERCAE)

Convolutional autoencoder works in combination to local values using simple convolution operation. This operation is used as encoder in convolutional autoencoder while its reverse operation is used as decoder to reconstruct original input. Its objective function is given as under:

$$\theta = y * log(y') - (1-y) * log\,(1-y') \quad (21)$$

where y' is basically predicted value. To this function as relation loss function following changes have been made.

$$\theta = (1-\alpha)\min L(y * logy' - (1-y) * log(1-y')) + \alpha \min L\left(\frac{\frac{\sum D^2}{N} - \left(\frac{\sum D}{N}\right)^2}{sum(D)}, \frac{\frac{\sum D'^2}{N} - \left(\frac{\sum D'}{N}\right)^2}{sum(D')}\right) \quad (22)$$

V. Implementation

This approach was implemented using open source library KERAS with Tensorflow as a backend. System used in this process is equipped with ci7 processor 7th generation with 16gb ram and NVIDIA 1050 ti GPU. Software toolchain used to implement the model consist of IPython development environment using Keras 2.0 on Tensorflow backend and nvidia cuda 8.0. Training and testing data is manipulated in form of numpy arrays.

5.1. Experiments

Experiments have been conducted on 2 most widely used datasets Cifar-10 and Mnist. Furthermore, results have been compared with state of the art methods [41-46], [47, p.], [48].

5.2. Datasets:

Above mentioned two datasets were divided into training, testing and validation sets. Description of these sets including dataset is given as follows:

CIFAR-10: This dataset has 10 categories having 32x32 color images. Each category has 6000 images, so, in total we have 60000 color images in this dataset. 50000 images were selected at random for training while remaining were divided in testing and validation at random.

MNIST: 10 categories of grayscale images having hand written images from 0 to 9 with total 70000 images of 28x28 pixels. Training was performed using 60000 images selected at random but equal from each class. Remaining 10000 were used for testing and validation purposes.

VI. Experimental Work

In this section model architecture, implementation details and training of these models on above mentioned datasets have been discussed.

6.1. Implemented Extended Relational Convolutional Autoencoder:

As main focus of this study is on objective function, so, we used standard structure of convolutional autoencoder with 3 conv2D layers in encoding process along with maxpooling layers. Four conv2D layers with upsampling have been used for decoding process. Structure of these layers with parameters and output shape is given in the table 1.

Adadelta was used as optimizer function. Adadelta dynamically adapts over time, using only first order information and produces minimal computational overhead without requiring manual tuning of learning rate and shows robustness to noisy gradient information, various data modalities, different model architectures and selection of hyper parameters. Adadelta is represented by eq. (27):

$$\Delta x_t = \frac{-RMS[\Delta x]_{t-1}}{RMS[g]_t} \quad (23)$$

where Δx_t is parameter update at time t and $RMS[g]_t$ is exponentially decaying average of RMS at t. This autoencoder was trained for 50 epochs with batch size of 256 with different loss functions to make a comparison.

Table 1 Structure of Extended Convolutional Autoencoder

Layer (type)	**Output Shape**	**Param #**
input_1 (InputLayer)	(None, 28, 28, 1)	0
conv2d_1 (Conv2D)	(None, 28, 28, 16)	160
max_pooling2d_1 (MaxPooling2)	(None, 14, 14, 16)	0
conv2d_2 (Conv2D)	(None, 14, 14, 8)	1160
max_pooling2d_2 (MaxPooling2)	(None, 7, 7, 8)	0
conv2d_3 (Conv2D)	(None, 7, 7, 8)	584
max_pooling2d_3 (MaxPooling2)	(None, 4, 4, 8)	0
conv2d_4 (Conv2D)	(None, 4, 4, 8)	584
up_sampling2d_1 (UpSampling2)	(None, 8, 8, 8)	0
conv2d_5 (Conv2D)	(None, 8, 8, 8)	584
up_sampling2d_2 (UpSampling2)	(None, 16, 16, 8)	0
conv2d_6 (Conv2D)	(None, 14, 14, 16)	1168
up_sampling2d_3 (UpSampling2)	(None, 28, 28, 16)	0
conv2d_7 (Conv2D)	(None, 28, 28, 1)	145

VII. RESULTS AND EVALUATIONS

In this section, we present the performance evaluations of the proposed objective function with different autoencoders in comparison to different loss functions for MNIST and CIFAR10 datasets. Different metrics have been used for evaluations and comparison of classification quality. These metrics include training accuracy, loss measurement, classification accuracy. Different loss functions have been used for comparison including Mean Square Error (MSE), Mean Square Log Error (MSLE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Relational Loss Function (RLF) and our proposed Extended Relational Loss Function (ERLF).

7.1. Training Accuracy with Convolutional Autoencoder:

Figures 1 and 2 show accuracy for MNIST and CIFAR10 datasets with Convolutional Autoencoder. Comparison have been made between different loss functions i.e. MSE, MSLE, MAE, MAPE, RLF with our proposed loss function. In these Figures it seems like our loss function is not having variations like others so Figures 3 and 4 clear this that variation is there and this loss function is working better than others for every epoch.

7.2. Loss with Convolutional Autoencoder:

Loss function tries to minimize the loss of data for better accuracy and retrieval. Training loss of Convolutional Autoencoder has been shown in Figures 5 and 6 with Mnist and Cifar10 datasets respectively. Though there is very small difference is there in comparison to some techniques but still ERLF is with lower loss and better accuracy as shown above.

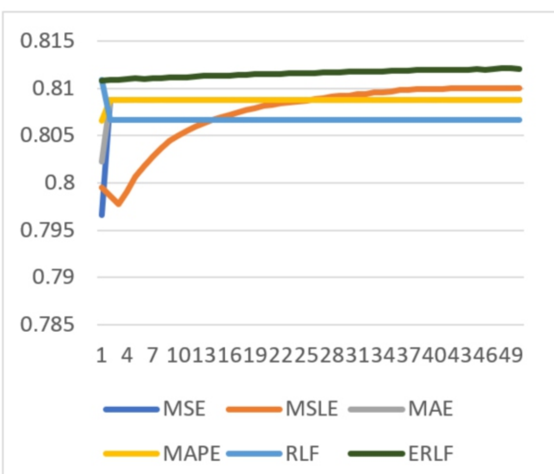


Figure 1: Training Accuracy for MNIST with Convolutional AE

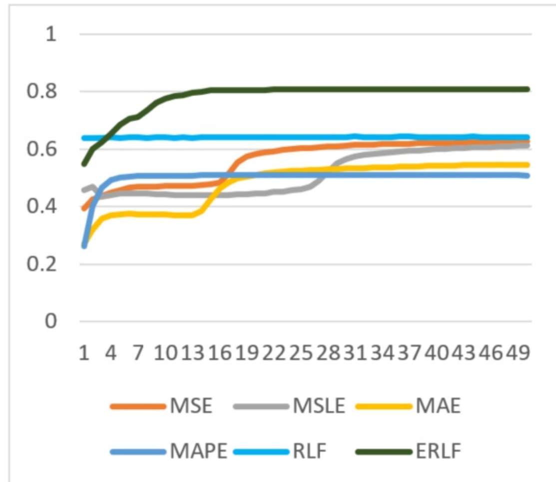


Figure 2: Training Accuracy for Cifar 10 with Convolutional AE

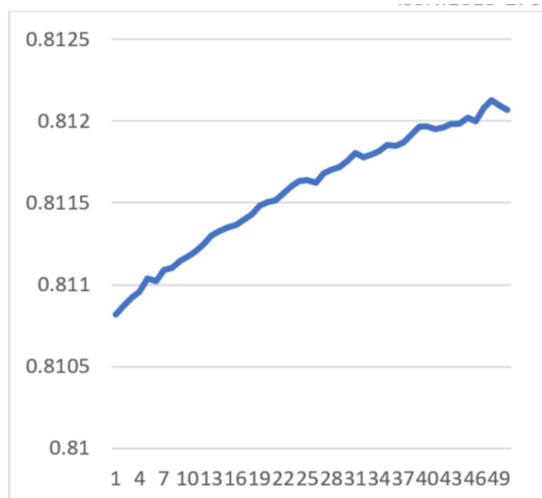


Figure 3: Accuracy of our Loss Function for MNIST

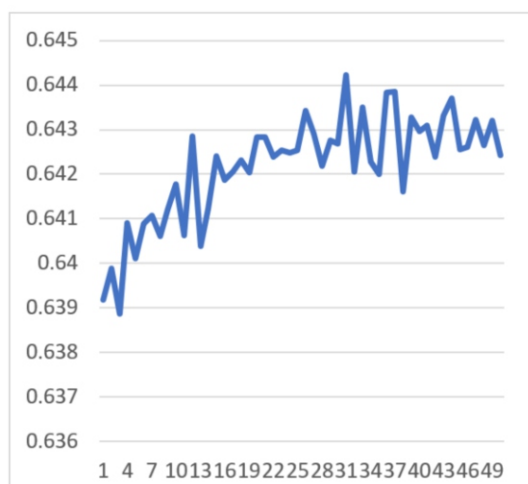


Figure 4 Accuracy of our Loss Function for Cifar10

7.3. Accuracies with Convolutional Autoencoder:

Extracted features were evaluated with state-of-the-art classification techniques i.e. K Nearest Neighbour (KNN), Decision Tree (DT), Random Forest (RF), Multilayer Perceptron (MLP), Adaboost (AB) and Quadratic Discriminant Analysis (QDA). In table 2, comparison of accuracies using above mentioned techniques with Mnist and Cifar 10 datasets have been shown. Results show that our proposed loss function ERLF is giving better accuracy in comparison to other loss functions.

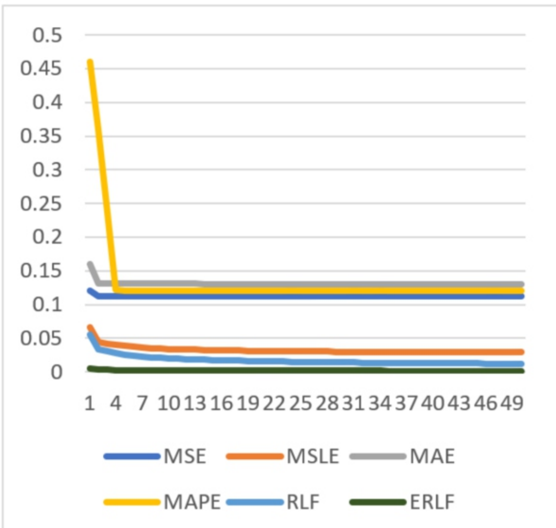


Figure 5: Training Loss for Mnist with Convolutional AE

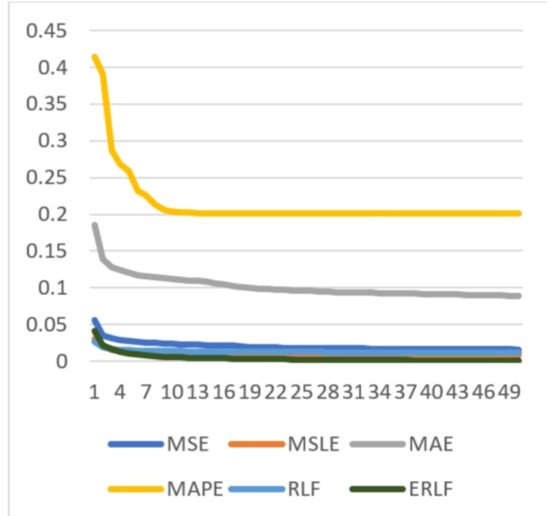


Figure 6: Training Loss for Cifar10 with Convolutional AE

VIII. GRAPHICAL RESULTS

Tables 3 and 4 show the graphical results with convolutional autoencoders using different loss functions. In these tables first row is showing original images while other rows are showing images reproduced from extracted features in encoding process. Results from ERLF are shown with better visual presentation for both Mnist and Cifar10. Again, one thing to mention here that these autoencoders were standard ones just to make comparison of loss functions. If some deep autoencoders are used with ERLF then we can have better accuracy results as well as outputs.

Table 2 Comparison of different classifiers with Convolutional Autoencoders

Algo	Mnist						Cifar10					
	ERLF	RLF	MSE	MSLE	MAE	MAPE	ERLF	RLF	MSE	MSLE	MAE	MAPE
KNN	0.853	0.803	0.794	0.814	0.77	0.775	0.842	0.801	0.786	0.782	0.789	0.716
DT	0.735	0.685	0.675	0.703	0.648	0.665	0.781	0.757	0.753	0.745	0.749	0.704
RF	0.826	0.782	0.777	0.797	0.757	0.765	0.778	0.769	0.755	0.659	0.659	0.695
MLP	0.868	0.752	0.762	0.819	0.756	0.816	0.768	0.707	0.735	0.688	0.693	0.692
AB	0.611	0.554	0.497	0.583	0.574	0.508	0.764	0.76	0.746	0.641	0.654	0.682
QDA	0.602	0.546	0.38	0.587	0.316	0.426	0.705	0.638	0.658	0.664	0.655	0.652

Table 3 Graphical Results with Mnist Dataset

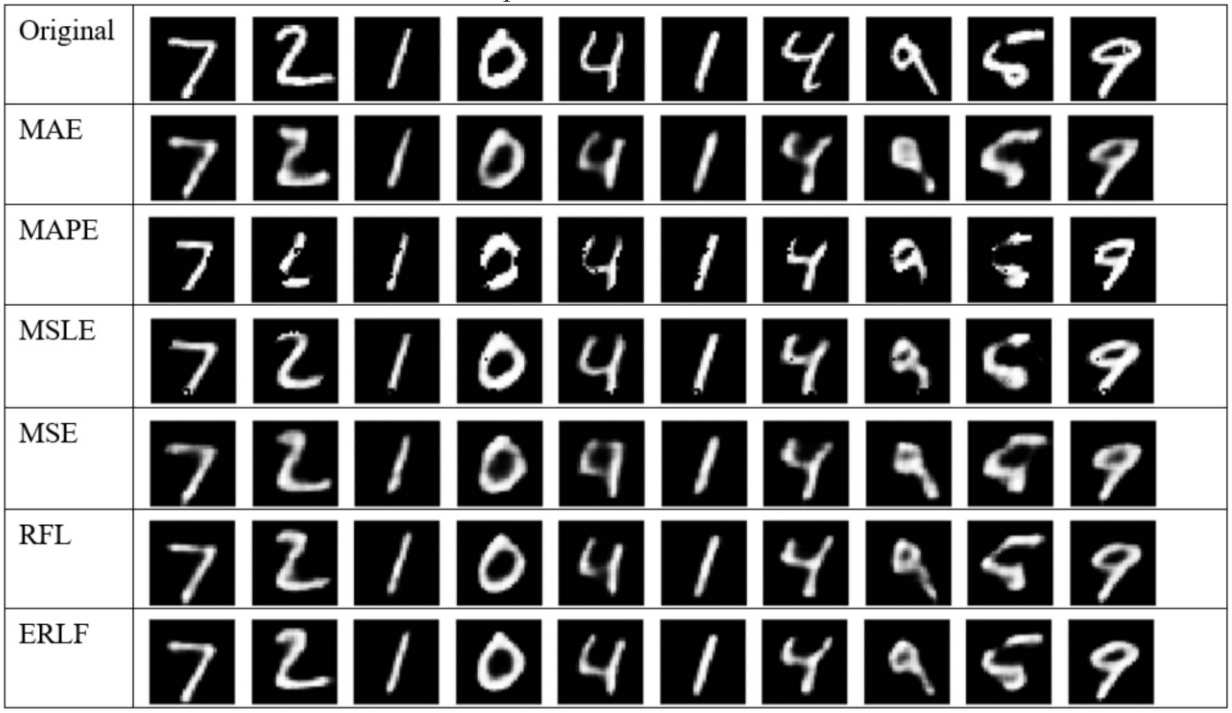
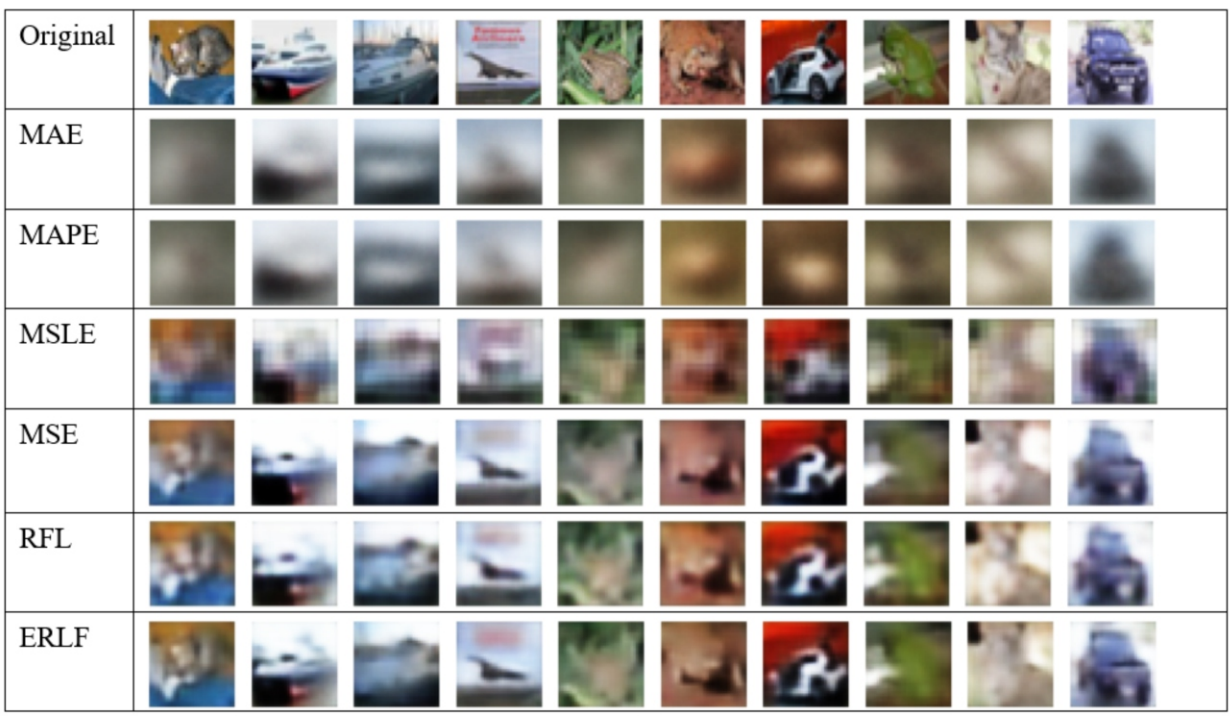


Table 4 Graphical Results with Cifar 10 Dataset



IX. Conclusion and Future Work

In this study we proposed a relational objective function that is helpful in high-level features extraction that use not only data but their relationship to extract these features. This function has been used with convolutional autoencoder. This autoencoder has been evaluated with different loss functions like Mean Square Error (MSE), Mean Square Log Error (MSLE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Relational Loss Function (RLF) and our proposed Extended Relational Loss Function (ERLF). Mnist and Cifar10 datasets have been used for evaluation and experimental results show that if we consider data along with its relationship, we can have better results with less loss. Extracted features will be more accurate and can contribute in better way for classification results improvement. For future, these extracted features can be used to retrieve some hash codes that can better represent an image and will be helpful for image retrieval. We use α with mean value by providing equal change to data and their relationship. This can also be tested with different values of α. This loss function can be tested on other available autoencoders to improve the performance.

References

[1] M. K. Bashir and Y. Saleem, "Deep Hashing for Semi-supervised Content Based Image Retrieval," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 8, pp. 3790–3803, Aug. 2018, Accessed: Nov. 11, 2020. [Online]. Available: http://www.itiis.org/digital-library/manuscript/2092.

[2] P. Indyk and R. Motwani, "Approximate nearest neighbors," 1998, doi: 10.1145/276698.276876.

[3] F. Wang and J. Sun, "Survey on distance metric learning and dimensionality reduction in data mining," *Data Mining and Knowledge Discovery*, vol. 29, no. 2, pp. 534–564, Jun. 2014, doi: 10.1007/s10618-014-0356-z.

[4] J. P. Cunningham and Z. Ghahramani, "Linear Dimensionality Reduction: Survey, Insights, and Generalizations," *Journal of Machine Learning Research*, vol. 16, pp. 2859–2900, 2015, [Online]. Available: http://jmlr.org/papers/v16/cunningham15a.html.

[5] N. Akkarapatty, A. Muralidharan, N. S. Raj, and V. P, "Dimensionality Reduction Techniques for Text Mining," in *Collaborative Filtering Using Data Mining and Analysis*, IGI Global, pp. 49–72.

[6] J. Wu, Z. Hong, S. Pan, X. Zhu, Z. Cai, and C. Zhang, "Multi-graph-view Learning for Graph Classification," in *2014 IEEE International Conference on Data Mining*, Dec. 2014, pp. 590–599, doi: 10.1109/ICDM.2014.97.

[7] J. Wu, Z. Cai, S. Zeng, and X. Zhu, "Artificial immune system for attribute weighted Naive Bayes classification," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, Aug. 2013, pp. 1–8, doi: 10.1109/IJCNN.2013.6706818.

[8] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant Features and the Subset Selection Problem," in *Machine Learning Proceedings 1994*, Elsevier, 1994, pp. 121–129.

[9] A. Liew, "Classification and Regression by RandomForest." https://www.researchgate.net/publication/228451484_Classification_and_Regression_by_RandomForest (accessed Aug. 23, 2018).

[10] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," Aug. 2014, doi: 10.1109/sai.2014.6918213.

[11] A. Sharma and K. K. Paliwal, "Linear discriminant analysis for the small sample size problem: an overview," *International Journal of Machine Learning and Cybernetics*, vol. 6, no. 3, pp. 443–454, Jan. 2014, doi: 10.1007/s13042-013-0226-9.

[12] U. Demšar, P. Harris, C. Brunsdon, A. S. Fotheringham, and S. McLoone, "Principal Component Analysis on Spatial Data: An Overview," *Annals of the Association of American Geographers*, vol. 103, no. 1, pp. 106–128, Jan. 2013, doi: 10.1080/00045608.2012.689236.

[13] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 1997, pp. 583–588.

[14] J.-M. Lee, C. Yoo, S. W. Choi, P. A. Vanrolleghem, and I.-B. Lee, "Nonlinear process monitoring using kernel principal component analysis," *Chemical Engineering Science*, vol. 59, no. 1, pp. 223–234, Jan. 2004, doi: 10.1016/j.ces.2003.09.012.

[15] P. Honeine, "Online Kernel Principal Component Analysis: A Reduced-Order Model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1814–1826, Sep. 2012, doi: 10.1109/tpami.2011.270.

[16] D. Lopez-Paz, S. Sra, A. J. Smola, Z. Ghahramani, and B. Schölkopf, "Randomized Nonlinear Component Analysis," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, Beijing, China, 2014, p. II–1359–II–1367, Accessed: Aug. 23, 2018. [Online]. Available: http://dl.acm.org/citation.cfm?id=3044805.3045044.

[17] *Multidimensional Scaling*. Routledge, 1987.

[18] J. B. Tenenbaum, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000, doi: 10.1126/science.290.5500.2319.

[19] S. T. Roweis, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000, doi: 10.1126/science.290.5500.2323.

[20] M. Belkin and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, Jun. 2003, doi: 10.1162/089976603321780317.

[21] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Error Propagation," Defense Technical Information Center, Sep. 1985. doi: 10.21236/ada164453.

[22] J. Deng, Z. Zhang, E. Marchi, and B. Schuller, "Sparse Autoencoder-Based Feature Transfer Learning for Speech Emotion Recognition," Sep. 2013, doi: 10.1109/acii.2013.90.

[23] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," 2008, doi: 10.1145/1390156.1390294.

[24] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 833–840, Accessed: Sep. 08, 2017. [Online]. Available: http://machinelearning.wustl.edu/mlpapers/paper_files/ICML2011Rifai_455.pdf.

[25] I. Goodfellow *et al.*, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680.

[26] W. Wang, Y. Huang, Y. Wang, and L. Wang, "Generalized Autoencoder: A Neural Network Framework for Dimensionality Reduction," Jun. 2014, doi: 10.1109/cvprw.2014.79.

[27] Q. Meng, D. Catchpoole, D. Skillicom, and P. J. Kennedy, "Relational autoencoder for feature extraction," in *2017 International Joint Conference on Neural Networks (IJCNN)*, May 2017, pp. 364–371, doi: 10.1109/IJCNN.2017.7965877.

[28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.

[29] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural Networks*, vol. 2, no. 1, pp. 53–58, Jan. 1989, doi: 10.1016/0893-6080(89)90014-2.

[30] N. Japkowicz, S. J. Hanson, and M. A. Gluck, "Nonlinear Autoassociation Is Not Equivalent to PCA," *Neural Computation*, vol. 12, no. 3, pp. 531–545, Mar. 2000, doi: 10.1162/089976600300015691.

[31] G. E. Hinton, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006, doi: 10.1126/science.1127647.

[32] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, "An empirical evaluation of deep architectures on problems with many factors of variation," 2007, doi: 10.1145/1273496.1273556.

[33] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," Sep. 2009, doi: 10.1109/iccv.2009.5459469.

[34] J.-L. Vincent, "International Study of the Prevalence and Outcomes of Infection in Intensive Care Units," *JAMA*, vol. 302, no. 21, p. 2323, Dec. 2009, doi: 10.1001/jama.2009.1754.

[35] W. W. Y. Ng, G. Zeng, J. Zhang, D. S. Yeung, and W. Pedrycz, "Dual autoencoders features for imbalance classification problem," *Pattern Recognition*, vol. 60, pp. 875–889, Dec. 2016, doi: 10.1016/j.patcog.2016.06.013.

[36] H.-C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach, "Stacked Autoencoders for Unsupervised Feature Learning and Multiple Organ Detection in a Pilot Study Using 4D Patient Data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1930–1943, Aug. 2013, doi: 10.1109/tpami.2012.277.

[37] M. aurelio Ranzato, C. Poultney, S. Chopra, and Y. L. Cun, "Efficient Learning of Sparse Representations with an Energy-Based Model," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. C. Platt, and T. Hoffman, Eds. MIT Press, 2007, pp. 1137–1144.

[38] I. Goodfellow, H. Lee, Q. V. Le, A. Saxe, and A. Y. Ng, "Measuring Invariances in Deep Networks," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 646–654.

[39] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations

in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010, Accessed: Sep. 07, 2017. [Online]. Available: http://www.jmlr.org/papers/v11/vincent10a.html.

[40] Brian Everitt, *The Cambridge dictionary of statistics*. Cambridge University Press, 1998.

[41] A. Gionis, P. Indyk, and R. Motwani, "Similarity Search in High Dimensions via Hashing," in *Proceedings of the 25th International Conference on Very Large Data Bases*, San Francisco, CA, USA, 1999, pp. 518–529, [Online]. Available: http://dl.acm.org/citation.cfm?id=645925.671516.

[42] Y. Weiss, A. Torralba, and R. Fergus, "Spectral Hashing," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. Curran Associates, Inc., 2009, pp. 1753–1760.

[43] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative Quantization: A Procrustean Approach to Learning Binary Codes for Large-Scale Image Retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, Dec. 2013, doi: 10.1109/TPAMI.2012.193.

[44] F. Shen, C. Shen, W. Liu, and H. T. Shen, "Supervised Discrete Hashing," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 37–45, doi: 10.1109/CVPR.2015.7298598.

[45] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised Hashing for Image Retrieval via Image Representation Learning," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, Qubec City, Qubec, Canada, 2014, pp. 2156–2162, [Online]. Available: http://dl.acm.org/citation.cfm?id=2892753.2892851.

[46] H. Lai, Y. Pan, Y. Liu, and S. Yan, "Simultaneous Feature Learning and Hash Coding with Deep Neural Networks," *CoRR*, vol. abs/1504.03410, 2015, [Online]. Available: http://arxiv.org/abs/1504.03410.

[47] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang, "Bit-Scalable Deep Hashing with Regularized Similarity Learning for Image Retrieval and Person Re-identification," *CoRR*, vol. abs/1508.04535, 2015, [Online]. Available: http://arxiv.org/abs/1508.04535.

[48] J. Zhang, Y. Peng, and J. Zhang, "SSDH: Semi-supervised Deep Hashing for Large Scale Image Retrieval," *CoRR*, vol. abs/1607.08477, 2016, [Online]. Available: http://arxiv.org/abs/1607.08477.

Paper Titled: Extended Relational Autoencoders for Feature Extraction in CBIR

Certificate

The subject article has not been published or submitted to or accepted for publication in any form, in any other journal or conference etc. Subject article is our own original contribution. We also guarantee that the authorship of this article will *not* be contested by anyone whose name(s) is/are not listed by us here and we will not request the authorities of journal to add any more author(s) after submission.

Corresponding Author

Authorship and Contribution Declaration

Sr.#	Author-s Full Name	Contribution to Paper	Signature
1	Dr. Muhammad Khawar Bashir	Proposed topic, basic study Design and methodology, manuscript writing	Signature by the Corresponding author on Behalf of Co-Authors
2	Dr. Sheraz Naseer	Data Collection, statistical analysis and interpretation of results etc.	
3	Dr. Yas ir Saleem	Literature review & Referencing, and quality insurer	