Deep learning based Image set classification for object recognition

Joshua Harvey, James Moodie, Christopher Gemmell, Nazier Roumani, Slade Lockyer and Milan Rosic

Abstract—***Machine learning models play an important role in the technological advancement of humanity as a whole. However, these models can be misled through the use of a noise, which can be hard to discern through human senses. There have been several approaches to defending against these nuisances. Our aim is to introduce a novel approach to defending against these nuisances, by employing Image Set Classification via Deep Learning. The employed Deep Learning architecture was a Deep Autoencoder to learn a meaningful lower dimensional representation of each image in a gallery set, and then reconstruct each image based on the lower dimensional representation. The error in image reconstrucion was then used to perform majority voting to assign a class label for the entire gallery set. We introduced noise in the training and testing of the Autoencoder structure via the Fast Gradient Sign Method. Our experimental results show that our Deep Image Set Classification model achieve an accuracy of 92% for object recognition task .***

Keywords—Image Set Classification; Deep Learning; Object recognition; Face recognition; Autoencoder, Fast Gradient Sign Method

I. INTRODUCTION

Szegedy [13] and their research team made a big discovery of machine learning models, which also included the neural network model [1]. Neural network models are a series of algorithms that recognize the relationships between different sets of data through a process that mimics a human brain, but these neural network models are quite vulnerable to the noisy data [1][2]. This means that these machine learning models are just slightly different than normal datasets.

There have been many types of research into finding solutions against these nuisances and the most common approach is the proper training of the model. Training the image to ignore the noise is the most basic type of strategy that there is. But even then, because there are so many different types of noise, this type of approach won't always work.

Hence, why we are trying to see whether training a set of images rather than just a single image will make a difference in the accuracy of the answers and give us a clear, true result. To test this, we will use the Deep Reconstruction Model for Image Set Classification that was developed by Hayat et al, which uses an autoencoder architecture and Gaussian Restricted Boltzmann Machines for pre-training. Section 2 will be a literature review which will provide you with the information of the Image Set Classification and will go into depth as to what Image Set Classification is. Section 3 will provide you with the implementation of our deep learning model. We will use the autoencoder structure that shows multiple equations to achieve good object recognition results. Section 4 and 5 will be the results that we have had during our testing period as well as the discussion of those results, that will go more into the depth of why or why not this method works.

II. LITERATURE REVIEW

Image set classification offers more promises than single image-based classification and has therefore attracted significant research attention in recent years [12]. Before we try to examine different image set classification techniques, we must first understand how image set classification works. The images which belong to the same class are viewed as an image set, then the most representative samples are extracted from the set, and finally, a proper model or probability distribution is learned to represent the intrinsic property of this set. The problem with image set classification is that images belonging to the same class can often be misrepresented based on key differences such as lighting as the issue is drawing intrinsic value from a set [4].

Image set classification methods can be divided into two types: parametric models and non-parametric. Parametric models solve image set classification problems by probability theories while non-parametric models aim to fit the distribution of samples by training the samples. Wu et al., [5] Proposed a Discriminant Tensor Dictionary Learning with Neighbour Uncorrelation (DTDLNU) model as dictionary-based models, although they can be effective, they transform each image set into a vector, which breaks the inherent spatial structure of the image set [6][7]. Zhao et al., [4] explored a method where a tensor is developed to model an image set with two spatial modes and one set mode, which fully explores the intrinsic structure of an image set. The authors claim that three challenging datasets were run and completed thus proving the effectiveness of DTDLNU.

Reverse training is a method proposed by [8] It begins by splitting training images from all classes into two sets labelled as 'D1' and 'D2'. D1 contains uniformly randomly sampled images from all classes with the total number of images in D1 being equal to the number of images in the query image set. The method was deemed efficient since it trains a single binary

classifier to optimally discriminate the class of the query image set from all others.

Stamatios [10] proposed a novel network architecture for grayscale and colour image denoising based on an iterative denoising scheme. The author acknowledges that the architecture of the proposed networks is more shallow than current deep CNN-based approaches however the resulting models lead to very competitive results particularly when the noise degrading the input deviates from the Gaussian assumption [10]. J. Lu, G. Wang, W. Deng, P. Moulin and J. Zhou [11] proposed a multi-manifold deep learning (MMDML) method for image set classification. Their method nonlinearly maps multiple sets of image instances into a shared feature subspace, so that discriminative, class-specific and nonlinear information are exploited for classification. J. Lu, [11] experimented on five popular subsets and claim that their method achieves better performance than the state-of-the-art image set classification techniques, however, they acknowledge that further testing is required in order to display the methods true potential.

Convolutional Neural Networks (CNN) are composed of several convolutional layers alternately connected with several pooling layers and can effectively characterize the essential features of the original image [9]. Guojian and Wenhui [9] used a convolution neural network structure which is a 6-layer structure, 4 layers are convoluted, and 2 layers are fully connected in order to classify different rock image sets. They warned that setting up an appropriate number of convolutional layers for the CNN is one of the most crucial steps. If there are too little the network will not be able to learn essential features of the original image and if there are too many it may lead to network over-fitting. The authors claim that their method has high accuracy when dealing with HSV, YCbCr or RGB colour space. The main concern is the experimental results that still have deviation which may be caused using single-polarized images. The authors warn that a few challenges arise when using CNN such as how many convolutional layers to use, the size of the convolutional kernel as well as the learning rate of the network and as a result are all worth to study carefully.

III. Methods and Experimental settings

A. Methods

To test a Deep Learning Image Set Classification model, we had used the model as developed by [14], a Deep autoencoder structure that employs unsupervised pre-training using Gaussian Restricted Boltzmann Machines to perform weight initialisation. This initialized Autoencoder is called the Template Deep Reconstruction Model (TDRM), the TDRM are then fine-tuned for each class k of the training image sets to produce a class-specific Deep Reconstruction Model (DRM). These tuned DRM's are used for Image Set Classification by employing a voting algorithm.

B. Autoencoder

The Template Deep Reconstruction Models are based on an autoencoder structure, which consists of two halves. The first half of the structure is an encoder with 3 fully connected layers, and the latter half is the decoder, also with three fully connected layers. The autoencoder structure has a shared layer at the centre. The encoder portion performs dimensionality reduction to find a meaningful representation of the input in lower dimensions, the latter decoder reconstructs the input image given the lower dimensional representation.

The encoder maps an input image $x \rightarrow h$ where x is the input image vector to the autoencoder, and h is the lower dimensional representation of the input, can be expressed as a combination of layers connected by some activation function (e.g. sigmoid, rectified linear unit).

$$h_1 = \sigma(W_e x + b_e)$$
$$h_2 = \sigma(W_e h_1 + b_e)$$
$$h = \sigma(W_e h_2 + b_e)$$

Fig. 1. Algorithm for the encoder portion of the autoencoder structure. The

Where σ represents the activation function, W_e represents the weight matrix for the encoder portion and b_e represents the biases for the encoder. The activations of each layer of the encoder structure are fed-forward to the next layer until the bottleneck of the autoencoder is reached, and the lower dimension representation has been discovered. The decoding process is like the encoding, where the decoder maps $h \rightarrow x'$, where x' is the reconstructed image. The decoder is again a combination of layers connected by some activation function, such that.

$$x'_1 = \sigma(W_d h + b_d)$$
$$x'_2 = \sigma(W_d x'_1 + b_d)$$
$$x' = \sigma(W_d x'_2 + b_d)$$

Fig. 2. Algorithm for the decoder portion of the autoencoder structure

Where W_d and b_d are the weight matrix and bias term respectively for the decoder portion of the autoencoder.

C. Initialisation and Training

With this, A complete autoencoder structure, or a Template Deep Reconstruction Model has been built. Thereafter, these Template Deep Reconstruction Models are fine-tuned for each class of an image set, producing class specific Deep Reconstruction Models. These Deep Reconstruction Models are trained to minimize a cost function using stochastic gradient descent and backpropagation. Tuning of the Template Deep Reconstruction Model may fail if the weight matrix is initialized inappropriately [16]. The weights of the TDRM are

therefore initialized with unsupervised pre-training using Gaussian Restricted Boltzmann Machines with a greedy layer-wise approach. After weight initialisation, class specific DRM's are trained to minimize the reconstruction error over all i training examples of a class.

$$J(TDRM) = \frac{1}{i}\sum_{j=1}^{i}\left\|x^{(j)} - x'^{(j)}\right\|^2$$

Fig. 3. The cost function is defined as the sum of the mean squared error of the input image against the reconstrcuted image

However, to avoid overfitting of the Deep Reconstruction Model, weight decay and sparsity target is included within the cost function as to introduce regularisation [16].

D. Image Set Classification

With the initialization and training of class-specific Deep Reconstruction Models, we can begin testing the network to reconstruct images from a test set, and then use a voting algorithm for classification of the set. The image set classification problem is defined as, given some test image set X, find the class y to which describes X. Image set classification using the tuned DRM's is as follows, for each image in a test set where $x \in X$, then for each class-specific Deep Reconstruction Model, perform a feed-forward pass on the current Deep Reconstruction Model using x as the input. Using the last layer of activations from the Deep Reconstruction Model, we can compute the reconstruction error for a class-specific DRM. With all the reconstruction errors covering all class-specific DRM's, a voting strategy can be applied to discover the class y for the test set X [16]. The voting strategy is as follows, each image x of the test set X casts only one vote. The vote x makes is directed towards the class whose corresponding class-specific Deep Reconstruction Model generates the lowest reconstruction error. After a vote has been cast for each image, the class that accumulates the most votes is deemed to be the class that represents the entire test set.

E. Adding Noise

For generation of noisy examples, the sign of the gradient for each image is computed and then multiplied by some small value epsilon to create the noise, this noise is then added back onto the clean image. The equation to calculate the noisy example is given:

$$x_{noise} = x + \varepsilon sign(\nabla_x J(TDRM))$$

Where x is the clean image, ε is some small value, and $\nabla_x J(TDRM)$ is the gradient of cost function with respect to the image x. The noise is added element-wise onto the clean image vector to produce x_{noise}.

F. Experimental settings

The training of this model provided and originally conducted by [16] created mini batches which consisted of 5 images and had an epoch value of 30. The model also used Majority Voting for classification and proved to have the second-best identification rate for the ETH-80 dataset according to the results in [16]. The initialization parameters and batch sizes of our experiment were not altered but the number of epochs for the training of the model was reduced to 20. The value of epsilon used in our experimentation were 0.1 and 0.25. These values were chosen empirically. The chosen amount of noisy batch percentage was chosen arbitrarily at 33% and 100% for image set classification.

IV. Results

Our experimentation used the ETH-80 Dataset [3] which allowed us to test the Template Deep Reconstruction Model provided by [16] for object recognition. This dataset consists of 8 object categories, these categories include apples, pears, tomatoes, cows, dogs, horses, cups, and cars. Of each of these 8 categories, there are 10 sub-objects which have large intra-class variations but still belong to that object category e.g. within the 'car' category exists 10 different car models, although each model has a large intra-class variation, it is still considered a car. For each of these objects, there are 41 separate images that belong to it, each image within this set has a different viewpoint situation above the object. Based on [16], each object of a certain category is considered as an image set. Each image of the ETH-80 set is 128 x 128 pixels, for use within the Deep Reconstruction Model each image is resized to 32 x 32 and converted from a three channel RGB into a single greyscale channel. Table 1 shows the mean accuracy and standard deviation of the Deep Reconstruction Model after four different runs of our experiments with different mixes of noisy batch percentage and epsilon value.

TABLE I. Results of noisy training and testing on ETH-80

Epsilon	**Percentage of noisy images per batch**	
	100%	***33%***
0.1	72 ± 16.52	92 ± 5.18
0.25	22 ± 10.36	91 ± 8.02

Fig. 4. The results gathered by generating noisy images within the training and test batches as per our experimental settings.

V. Discussion

Hayat, Bennamoun and An [16] had shown with their Deep Reconstruction Model, state of the art performance in the image set classification task when compared to non-deep learning image set classification techniques. Hayet et al,. [16] had achieved a classification accuracy of 98.21 ± 1.69 when running experimentation on the ETH-80 dataset. However, Figure 1 exhibits that when the Deep Reconstruction Model is

exposed to noisy data, classification accuracy begins to fall. Firstly, when only 33% of both the training and testing batches are noisy, and the epsilon value is at $\varepsilon = 0.1$, accuracy decreases to 92 ± 5.18. Moreover, when epsilon is $\varepsilon = 0.25$ at the same noisy patch percentage, accuracy falls further to 91 ± 8.02. Furthermore, when increasing the noise to 100% for both testing and training, accuracy falls even further to 72 ± 16.52 and 22 ± 10.36 for $\varepsilon = 0.1$ and $\varepsilon = 0.25$ respectively. When testing and training batches are only 33% noisy, the difference in classification accuracy is somewhat comparable to the state-of-the-art accuracy displayed in [15], however, the standard deviation has risen by a significant amount. Furthermore, at 100% noisy batch the accuracy of the Deep Reconstruction Model has dropped by a significant amount for both tested values of epsilon.

We now discuss our results in relation to noise on single image classification models. Chen and Sirkeci-Mergen [16] found that using a basic autoencoder had less of an improvement on their model's accuracy compared to when the denoising autoencoder was used when the noise value was increased. The results that were gathered in [16] regarding the denoising autoencoder showed an accuracy of 93.57% with an epsilon value of 0.25 on noisy examples whereas when the epsilon value was 0.1, the values in the graph showed a greater accuracy. Although we cannot compare the results found in [16] to our results as their model was used for single image classification, we can see from the difference between epsilon values of 0.1 and 0.25 that the accuracy decreases as the amount to noise the images is increased.

From these results, and compared to [16], Image Set Classification has shown to be somewhat effective. The inconsistency in the data results indicates that further tests need to be carried out over a longer period. Testing on noisy data was only done for object recognition using the ETH-80 Dataset which meant that our findings are only applicable for object recognition with this dataset.

VI. Conclusion

The use of a neural network model increasingly in everyday life has progressed the need for identification tasks. In our paper, we used the model previously mentioned to test how noise affect object recognition accuracy. We found that when our model is tested, accuracy fell even for only 33% noisy test data. When it was increased to 100%, accuracy fell much more substantially. Recognizing image set classification models in our literature review allowed us to generate a deeper understanding of neural networks and these techniques. Whilst we mainly focused on the FGS method, there are many more forms of nuisances, which can be added to images. We did experience some limitations throughout the experiment, for example, if the experimentation time had been extended, we would have suggested to test the accuracy of this model with additional datasets such as the Mobo or Kinect dataset as this TDRM model showed high accuracy in the results of [1] for face and text recognition.

References

[1] C. Szegedy et al., "Intriguing properties of neural networks", 2013. [Accessed 25 May 2019].

[2] J. Chen, "Neural Network Definition", *Investopedia*, 2019. [Online]. Available: https://www.investopedia.com/terms/n/neuralnetwork.asp. [Accessed: 25 May 2019].

[3] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization", *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*. Available: 10.1109/cvpr.2003.1211497 [Accessed 25 May 2019].

[4] Z. Zhao, S. Xu, D. Liu, W. Tian and Z. Jiang, "A review of image set classification", *Neurocomputing*, vol. 335, pp. 251-260, 2019. Available: 10.1016/j.neucom.2018.09.090.

[5] F. Wu, X. Jing, W. Zuo, R. Wang and X. Zhu, "Discriminant tensor dictionary learning with neighbor uncorrelation for image set based classification", *International Joint Conference On Artificial Intelligence*, 2017.

[6] "Image Set-Based Collaborative Representation for Face Recognition", *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 7, pp. 1120-1132, 2014. Available: 10.1109/tifs.2014.2324277.

[7] Z. Chen, B. Jiang, J. Tang and B. Luo, "Image Set Representation and Classification with Attributed Covariate-Relation Graph Model and Graph Sparse Representation Classification", *Neurocomputing*, vol. 226, pp. 262-268, 2017. Available: 10.1016/j.neucom.2016.12.004.

[8] M. Hayat, M. Bennamoun and S. An, "Reverse Training: An Efficient Approach for Image Set Classification", *Computer Vision – ECCV 2014*, pp. 784-799, 2014. Available: 10.1007/978-3-319-10599-4_50.

[9] G. Cheng and W. Guo, "Rock images classification by using deep convolution neural network", Journal of Physics: Conference Series, vol. 887, p. 012089, 2017. Available: 10.1088/1742-6596/887/1/012089.

[10] S. Lefkimmiatis, "Universal Denoising Networks : A Novel CNN Architecture for Image Denoising", *Skolkovo Institute of Science and Technology (Skoltech), Moscow, Russia*, 2018.

[11] J. Lu, G. Wang, W. Deng, P. Moulin and J. Zhou, "Multi-Manifold Deep Metric Learning for Image Set Classification", *School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore*, 2015.

[12] M. Hayat, M. Bennamoun and S. An, "Deep Reconstruction Models for Image Set Classification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 4, pp. 713-727, 2015. Available: 10.1109/tpami.2014.2353635 [Accessed 11 May 2019].

[13]. C. Szegedy et al., Intriguing properties of neural networks, 2014, [online] Available: https://arxiv.org/abs/1312.6199.

[14] M. Hayat, M. Bennamoun and S. An, "Deep Reconstruction Models for Image Set Classification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 4, pp. 713-727, 2015. Available: 10.1109/tpami.2014.2353635 [Accessed 25 May 2019].

[15] G. Hinton, S. Osindero and Y. Teh, "A fast learning algorithm for deep belief nets." Available: https://www.cs.toronto.edu/~hinton/absps/fastnc.pdf. [Accessed 25 May 2019].

[16] I. Chen and B. Sirkeci-Mergen, "A Comparative Study of Autoencoders against Adversarial Attacks", 2018. Available:

https://csce.ucmss.com/cr/books/2018/LFS/CSREA2018/IPC3651.pdf.
[Accessed 25 May 2019].