Deep Learning Methods for Classification with Limited Datasets

Ariana Familiar & Roshan Santhosh

Abstract

The success of deep convolutional neural networks for image classification tasks often relies on extremely large datasets for model training. However, large amounts of data are not always available, and training on large datasets is computationally expensive and inefficient. In this project, we explore the use of Siamese neural networks for image classification with a limited dataset. We compared Siamese network performance with conventional CNNs on two tasks, mathematical digit and face recognition, and found the Siamese network had higher classification performance and required fewer training epochs compared to the CNNs for both tasks. Overall, the Siamese network proved better than the CNN for classification given small training dataset sizes and significant image-based variance within classes. These results show the strength of Siamese neural networks for images classification with limited datasets.

1. Introduction

Machine learning (ML) techniques have significantly advanced the performance of classification tasks by computers. The use of convolutional neural networks (CNNs), and particularly deep convolutional networks trained on large image datasets has been utilized across a wide variety of applications. However, serious challenges remain in implementing classification algorithms at smaller scales.

Success in the use of deep learning architecture for classification is typically associated with extremely large datasets for model tuning, on the order of millions of images (e.g., Schroff, Kalenichenko, & Philbin, 2015). This can be computationally expensive and inefficient, as well as potentially impossible when only small datasets are available.

On the other hand, neural networks based on similarity metrics have proven successful for classification based on limited data. Siamese neural networks, in particular, involve a contrastive loss function that is used to compute similarity between images during training. Siamese networks have been shown to achieve one-shot image recognition (Koch, Zemel, & Salakhutdinov, 2015) as well as human-level performance for face verification (Taigman et al., 2014).

Our focus for this project is to explore the application of Siamese Networks in image classification tasks with less amounts of training data and to compare its performance against conventional CNNs. In this regard, we have selected two different domains for testing Siamese Networks:

- Digit Recognition
- Face Recognition

For the digit recognition task, we compared Siamese network and CNN model performance across a range of training set sizes (5, 10, 30, 100, 300, 1000 images per class).

For the face recognition task, we compared Siamese network and CNN performance for a small dataset of face images in which there was significant variance in image properties (illumination, viewpoint, facial expression, facial details (e.g. glasses, facial hair)) within and across people.

We found that the Siamese network outperformed conventional CNNs for both digit and face recognition tasks. This was true for performance (classification accuracy) and efficiency (number of epochs required for training) metrics.

2. Siamese Networks

Siamese Networks are a modification over conventional Convolutional Neural Networks that allow for the training of classification models with less amount of data for training. The focus of the model is to be able to differentiate images from different classes rather than the accurate prediction of class for each image.

This goal is achieved through the use of a special loss function known as the **Contrastive loss** function. Contrastive loss function is ideal for the network and its task as it forces vector representations of images from the same class to be as similar to each other, while at the same time making vector representation of images from different classes are as dissimilar as possible. Comparisons can be drawn to the creation of word vectors in the word2Vec algorithm.

2.1 Model Architecture

The model consists of 2 pathways, consisting of CNN model layers, that take in two images at a time (Figure 1). These pathways are identical in all aspects and even share the same weights throughout training and testing. The output from both these pathways would be vector representations of the images that were passed as input to them. These vector representations are then passed through the contrastive loss function to assess the distance between them.

2.2 Model Input

Given the model architecture, the input data needs to be provided in a particular format, specifically as pairs. The data preprocessing includes the additional task of creating pairs of similar/dissimilar images. Images from the same class are taken to create similar pairs while images from different classes are taken to create dissimilar pairs. Similar pairs are given a label of 1 while dissimilar pairs are given a label of 0.

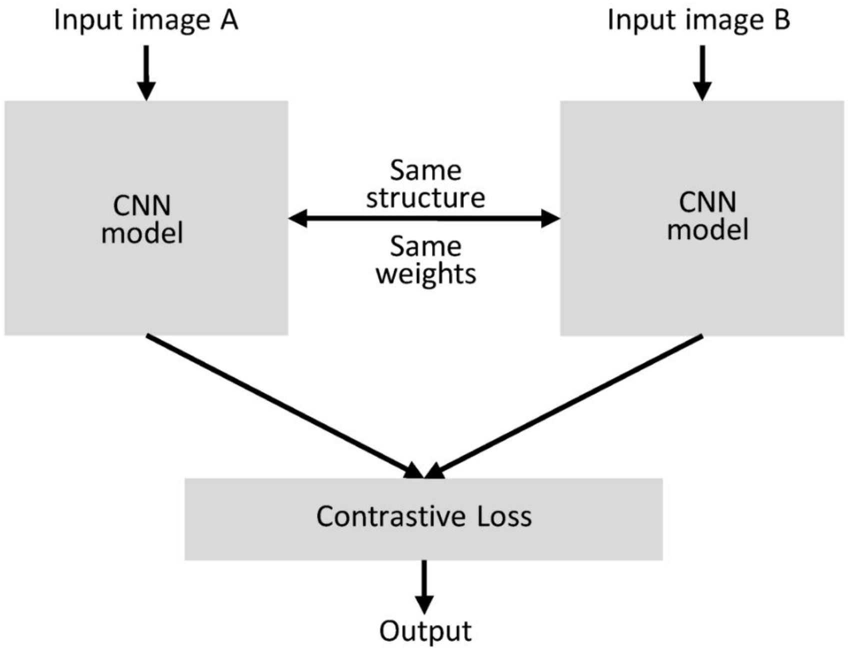

Figure 1. Depiction of Siamese Neural Network architecture.

2.3 Contrastive Loss Function

Contrastive Loss is often used in image retrieval tasks to learn discriminative features for images. During training, an image pair is fed into the model with their ground truth relationship y : y equals 1 if the two images are similar and 0 otherwise.

$$yd^2 + (1 - y)\max(margin - d, 0)^2$$

where d is the Euclidean distance between the two image features. The margin term is used to "tighten" the constraint: if two images in a pair are dissimilar, then their distance should be at least margin , or a loss will be incurred.

3. Convolutional Neural Networks

We used Keras with a TensorFlow backend to implement conventional CNNs as a benchmark for assessing the performance of the Siamese network. For the face recognition task, we used two CNN architectures, one we are calling “CNN-4” that has been shown to have successful performance for image classification based on large datasets, and another we are calling “CNN-11” that has been shown to have successful performance for face recognition given variance in illumination and viewpoint (Schroff, Kalenichenko, & Philbin, 2015). For the digit recognition task, we used only CNN-4.

3.1 Model Architecture & Tuning

The architecture of the first CNN ("CNN-4") is outlined in Table 1. There were four convolutional layers with max-pooling followed by three densely connected layers and one softmax activation layer.

	Filters	Size	Stride
conv	32	3 x 3	3
pool		2 x 2	2
conv	64	3 x 3	3
pool		2 x 2	2
conv	128	3 x 3	3
pool		2 x 2	2
conv	256	3 x 3	3
pool		2 x 2	2
flatten			
full connected	256		
dropout (50%)			
full connected	256		
dropout (50%)			
full connected	40		
softmax			

Table 1. CNN-4 architecture. All convolutional layers were followed by ReLUs).

The architecture of the second CNN ("CNN-11") is outlined in Table 2. There were eleven convolutional layers with max-pooling followed by three densely connected layers and one softmax activation layer.

	Filters	Size	Stride
conv	64	7 x 7	2
pool		2 x 2	2
norm			
conv	64	1 x 1	1
conv	192	3 x 3	1
norm			
pool		2 x 2	2
conv	192	1 x 1	1
conv	384	3 x 3	1
pool		2 x 2	2
conv	384	1 x 1	1
conv	256	3 x 3	1
conv	256	1 x 1	1
conv	256	3 x 3	1
conv	256	1 x 1	1
conv	256	3 x 3	1
pool		2 x 2	2
flatten			
dense connected	128		
dropout (50%)			
dense connected	128		
dropout (50%)			
dense connected	40		
softmax			

Table 2. CNN-11 architecture. All convolutional layers were followed by ReLUs).

In both CNNs, all convolutional layers were followed by rectified linear units (ReLUs) and dropout was used after the fully connected layers in order to reduce model over-fitting.

For tuning, a cross entropy loss function and a learning rate of 0.00001 were used. For the digit recognition task, a decay rate of 1e-6 was used. All models were trained until convergence, and the number of epochs required depended on the training set size.

4. Digit Recognition

4.1 Dataset

The MNIST database of handwritten mathematical digits (LeCun, Bottou, Bengio, & Haffner, 1998) was retrieved via Keras, which consists of unique images for each 0-9 digit. All images were greyscale, 28x28 pixels in size (see Fig. 2 for exemplars).

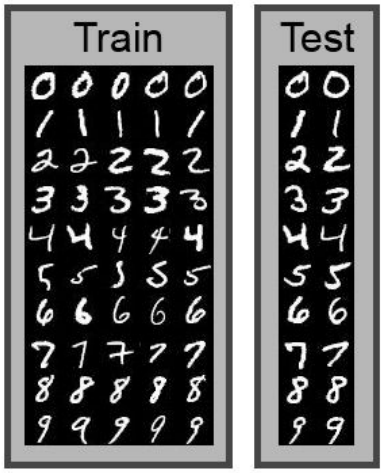

Figure 2. Examples of training and testing images from the MNIST dataset.

The dataset consists of 60,000 images in the training dataset and 10,000 images in the test dataset. For our models, we randomly selected fixed number of images from each class for training. The training set sizes that were used for training were: 5, 10, 30, 100, 300, 1000. For testing, 100 randomly selected images from each class were selected and predicted for using the models. Training and testing sets were independent.

4.2 Model Architecture and Tuning

For the CNN component of the Siamese Net, we chose the following architecture for the network:

Layer	Filters	Size	Stride
Conv	64	3x3	1
Conv	64	3x3	1
Max pool		2x2	1
Dropout			1
Conv	128	3x3	
Dropout			
Conv	256	3x3	
Conv	256	3x3	
Max Pool		2x2	
Dropout			
Flatten			
Dense	256		

Table 3. CNN architecture within the Siamese network for digit recognition task

The model used Euclidean distance as the distance metric to measure the difference between the two input image vectors. Contrastive loss function was used with a margin value of 5. The optimization was done using Adadelta optimizer. Since Adadelta was used, a learning rate was not specified. However, learning rate was adjusted to reduce by a factor of 0.1 after every 3 consecutive epochs with no improvement in validation data loss.

4.3 Model Performance Evaluation

For most sizes of the training data sets, the Siamese Net outperforms the CNN model (with the exception of equivalent performance at 10 training images per class; Fig. 3). The difference is much more significant when the training set size is >= 30. The lack of significant difference in results between the models with really low training set size (10 images) is likely because of the lack of variability in the really small training set. The training of the Siamese Net is also faster as all Siamese network models were trained within 20 epochs.

Overall, the Siamese network had a much steeper learning curve, in that it achieved high levels of classification at smaller training set sizes compared to the CNN. This shows how the Siamese network outperforms the CNN in the digit recognition task, and how the Siamese network has high classification even at very small training set sizes.

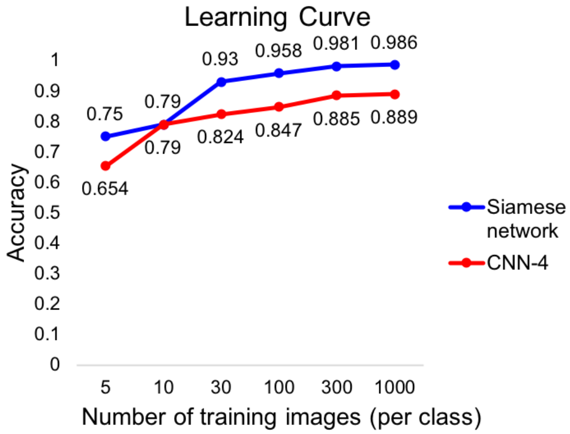

Figure 3. Classification accuracy for digit recognition task. Results are shown for different numbers of training images per digit class.

5. Face Recognition

5.1 Dataset

Images were retrieved from the Our Database of Faces from AT&T Laboratories (http://www.cl.cam.ac.uk/Research/DTG/attarchive:pub/data/att_faces.tar.Z). There were 10 images of each of 40 distinct people. For some people, there were differences in illumination, viewpoint, facial expressions (open/closed eyes, smiling/not smiling), and facial details (glasses/no glasses; see Figure 4 for exemplars). All images were taken against a dark, homogeneous background, and were 92 x 112 pixels in size with 256 gray levels per pixel.

Images were converted from PGM to JPG. The following image preprocessing was performed in Keras: rescaling (dividing pixel value by 255) and random transformations of training images (shearing (angle of 0.1 degree), horizontal flipping).

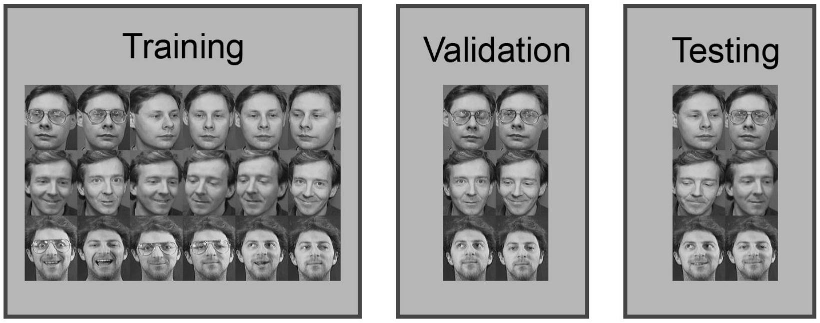

Figure 4. Examples of images in the dataset. Each row corresponds to one class (face identity). All images were 92 x 112 pixels and grayscale. Training, validation, and testing image sets were independent.

Images were split such that for each face identity there were 6 images for training, 2 images for validation, and 2 images for testing. The images in each of these subsets were independent (no image was repeated across subsets).

5.2 Model Architecture and Tuning

The same architecture and tuning parameters were used as described in the digit recognition section (4.2).

5.3 Model Performance Evaluation

For the face recognition task, the Siamese network had 100% classification performance on the test set. This was higher than the 97.5% classification accuracy for both CNN-4 and CNN-11 models. The training was also significantly faster for the Siamese network compared to both CNNs, with only 20 epochs required for training the Siamese network, 700 epochs for the CNN-4, and 1100 epochs for the CNN-11 models.

	Accuracy
Siamese network	1.0
CNN-4	0.975
CNN-11	0.975

Table 5. Results for the face recognition task (classification accuracy of the test set).

6. Conclusions

In this project, we compared the performance of a Siamese neural network with conventional CNNs on two separate tasks. The first task was recognition of handwritten digits, and the Siamese network had higher classification performance compared to a CNN with four convolutional layers. We evaluated performance of each model for a range of training set sizes (5, 10, 30, 100, 300, 1000 images per digit class), and found that the Siamese network had a steeper learning curve compared to the CNN, and had high classification performance even at the smallest training set sizes used.

The second task was recognition of face identities using a small dataset of images (10 total images per person) with significant image-based variance. Both within and between people there were differences in illumination, facial expression, viewpoint, and facial details (e.g. glasses, facial hair). Again, we found that the Siamese network had higher performance in recognizing faces compared to two CNNs (one with four convolutional layers, one with eleven).

Additionally, for both tasks the Siamese network required smaller numbers of epochs for model training. Thus, by both classification performance and efficiency metrics, the Siamese network proved better than conventional CNNs.

Overall, the Siamese network performed better than the CNN with limited datasets (both in the small sizes of the training set and image-based variance). As the Siamese network architecture involves comparisons of input pairs, and uses a similarity metric (in our case, Euclidean distance) to perform such comparisons, it is well-suited for classification tasks such as the ones in this project that involved image comparisons.

In particular, the strength of the Siamese architecture is twofold. First, the parallel CNNs share weights, which results in fewer parameters to train and thus there is less data required for training and a lower susceptibility to overfitting (and consequently, better generalization to separate datasets). Second, each CNN within the Siamese network produces a representation of its input and the contrastive loss function is used to differentiate between representations of input pairs by computing their similarity. As a result, it can be less susceptible to image-based variance for classification tasks, provided there is comparable variance in the training and testing datasets.

In sum, we found that the Siamese neural network outperformed conventional CNNs for digit and face recognition tasks. The use of Siamese networks is appropriate for cases in which there is a limited amount of data available for training. This is beneficial for situations in which collecting large datasets for model training is difficult. Additionally such architectures could prove useful for building artificial intelligence systems with human-level performance (as has been shown for one-shot learning of images, Koch, Zemel, & Salakhutdinov).

7. References

Koch, G., Zemel, R., & Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop* (Vol. 2).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 815-823).

Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1701-1708).