Hand Written Characters Recognition via Deep Metric Learning

Shah Nawaz[1], Alessandro Calefati[1], Nisar Ahmed[2], Ignazio Gallo[1]
[1]University of Insubria, Italy
[2]University of Engineering and Technology, Lahore Pakistan
snawaz@uninsubria.it, a.calefati@uninsubria.it, 2015phdce02@student.uet.edu.pk, ignazio.gallo@uninsubria.it

***Abstract*—Deep metric learning plays an important role in measuring similarity through distance metrics among arbitrary group of data. MNIST dataset is typically used to measure similarity however this dataset has few seemingly similar classes, making it less effective for deep metric learning methods. In this paper, we created a new handwritten dataset named Urdu-Characters with set of classes suitable for deep metric learning. With this work, we compare the performance of two state-of-the-art deep metric learning methods i.e. Siamese and Triplet network. We show that a Triplet network is more powerful than a Siamese network. In addition, we show that the performance of a Triplet or Siamese network can be improved using most powerful underlying Convolutional Neural Network architectures.**

***Keywords*-Handwritten dataset; Deep metric learning; Triplet network; Siamese network;**

I. Introduction

The aim of deep metric learning is to learn a similarity metric from data. The similarity metric can be used later to compare or match new samples from previously unseen data. In recent years, deep metric learning has gained considerable popularity following the success in deep learning. Deep metric learning can be applied to numerous applications such as retrieval [1], [2], clustering [3], feature matching [4], verification [5], [6] etc. Extreme classification [7], [8] with enormous number of classes can also take advantage of deep metric learning methods because of its ability to learn the general concept of distance metrics.

Typically, deep metric learning methods are built on underlying state-of-the-art Convolutional Neural Networks (CNNs) [9], [10], [11]. Deep metric learning methods produce an embedding of each input so that a certain loss, related to distance between two images, is minimized. In other words, embedding produced by metric learning methods are optimized to push examples of similar classes closer, conversely examples belonging to different classes are far from them. Such embedding is robust against intra-class variation which makes such methods suitable to learn similarity. Existing works take randomly sampled pairs of similar and dissimilar inputs or triplets consisting of query, positive and negative inputs to compute the loss on individual pairs or triplets.

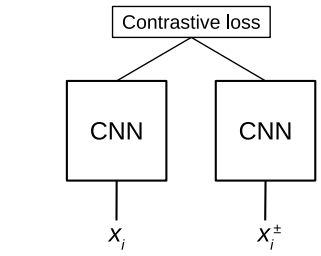
Computer vision community has extensively used MNIST dataset in different applications including similarity. However, the dataset has only few seemingly similar classes, making it less effective for deep metric learning methods. In this paper, a new handwritten dataset named Urdu-Characters is created in a similar way as MNIST dataset. Furthermore, we build Siamese and Triplet networks on Urdu-Characters and MNIST datasets to show that a Triplet network is more powerful than a Siamese network. We demonstrated that the performance of a Siamese or Triplet network can be improved further using most powerful underlying Convolutional Neural Network architectures i.e. Alexnet [10] and Googlenet [9].

II. Related Work

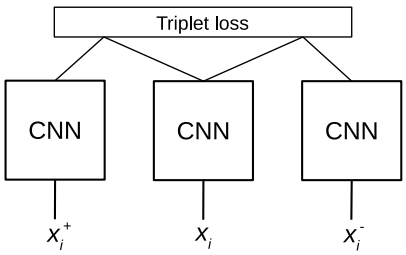
Kulis [12] provides a comprehensive survey on advances in metric learning. Siamese models have been used for very different purposes. For example Bromley *et al.* [13] presented a Siamese network for signature verification, while Chopra *et al.* [5] used a similar network for face verification. They pointed out a complete freedom in the choice of underlying architecture to build such family of networks. This observation is extremely important as future variants of Siamese network are built on top of more powerful architectures i.e. Alexnet [10], Googlenet [9] etc.

With rise in e-commerce websites, deep metric learning methods are extensively employed in image retrieval applications, for example Bell and Bala [14] used variants of Siamese network to learn an embedding for visual search in an interior design context. The embedding produced by such network is then used to search for products in the same category, searching across categories and looking for a product in an interior scene. They concluded that using higher dimension of the embedding makes it easier to satisfy constraints in loss function. However, higher embedding dimension will significantly increase the amount of space and time required to search image in retrieval applications. In our work, we show that higher embedding dimension produces better results for such networks. However, the choice of embedding dimension is based on the application context. Veit *et al.* [2] extended the Siamese network to answer this question: *'What outfit goes well with this pair of shoes?'*. The proposed framework learns compatibility between items from different categories consisting of outfit and shoes. In other words, it goes beyond the notion of similarity using the notion of style. Their work is also one of the interesting applications of a Siamese network. Wang *et al.* [15] presented deep ranking model to learn fine-grained image similarity models based on triplet loss. Schroff *et al.* [6] used the similar loss for face verification, recognition and clustering. Authors also presented an online triplet mining method for creation of triplets. Similarly, we perform experiments with three triplet sampling strategies to an-

978-1-5386-3346-5/18 $31.00 © 2018 IEEE
DOI 10.1109/DAS.2018.18



(a) Siamese network



(b) Triplet network

Figure 1: Graphical representation of networks with Contrastive and Triplet loss functions used in this work.

alyze the impact of triplet creation on the network. In this work, we analyze the performance of Siamese and Triplet networks with different underlying CNN architectures. We also analyzed the impact of embedding dimensionality on these deep metric learning methods. These two aspects of our work are not deeply explored in related works. Typically, deep metric learning methods used MNIST dataset in experiments. However, with the introduction of Urdu-Characters dataset, we provide researchers with a dataset with higher number of classes and ambiguities among classes. The nature of this dataset can be ideal for deep metric learning and classification tasks.

III. Deep Metric Learning Methods

A Siamese or Triplet network learns distance metric where similar examples are mapped close to each other and dissimilar examples are mapped farther apart.

A. Siamese Network

Siamese network shown in Figure 1a is popular among tasks that involve finding similarity or a relationship between two comparable things. The network is characterized by using the contrastive loss function during the training which pulls together items of a similar class while pushing apart items of different classes. The formula is shown below:

$$L_s(x_i, x_i^{\pm}) = \sum_{i}^{N}[(1-y)\|f(x_i) - f(x_i^{\pm})\|_2^2 + \\ +y \cdot max(0, \alpha - \|f(x_i) - f(x_i^{\pm})\|_2^2)] \quad (1)$$

where N stands for the number of images in the batch, $f(\cdot)$ is the feature embedding output from the network, $\|f(x_i) - f(x_i^{\pm})\|_2^2$ is the Euclidean distance to measure the similarity of extracted features from two images, and the label $y \in \{0, 1\}$ indicates whether a pair $(x_i, x_i^{\pm})$ is from the same class or not.

The training process for this kind of network is done feeding a pair of images x_i, $x_i^{\pm}$ and a label $y \in \{0, 1\}$ representing the similarity or dissimilarity between images.

B. Triplet network

The Triplet network (Figure 1b) is an extension of the Siamese network. It consists of three instances of the same feed-forward network (with shared parameters). The triplet loss [6] is trained on a series of triplets $\{x_i, x_i^+, x_i^-\}$, where x_i and x_i^+ are images from the same class, and x_i^- is from a different class, as reported in Equation 2. The triplet loss is formulated as following:

$$L_t(x_i, x_i^-, x_i^+) = \sum_{i}^{N} max(0, [\|f(x_i) - f(x_i^+)\|_2^2 - \\ -\|f(x_i) - f(x_i^-)\|_2^2] + \alpha]) \quad (2)$$

where $f(x_i), f(x_i^+), f(x_i^-)$ mean features of three input images and α is a margin that is enforced between positive and negative pairs.

C. Covolutional Neural Networks

A Siamese or Triplet network is built on top of underlying CNN architecture as shown in Figure 1. A full description of CNN is beyond the scope of this paper; however we present a brief overview of CNN. Typically, a CNN structure consists of various stages or layers such convolutional, pooling and rectification. Parameters in each layer are learned from training data to optimize performance on some tasks. Alexnet [10] is considered first CNN model successfully applied for image classification and starting from this architecture many new architectures have been presented in recent years. In our work, we use three well-known CNNs (Lenet, Alexnet and Googlenet), as underlying architectures to build a Siamese network or Triplet network. Lenet has only two convolutional layers while AlexNet has 5 convolutional layers and Googlenet has many more layers. It is important to note that 'softmax' layer is removed from these architectures to obtain D-dimensional embedding.

D. Triplet Sampling

We employ three different strategies for triplet creation in our experiments. We want to evaluate if the creation process has an impact on the overall performance. The first strategy that we employ consists of random selection of an image from the dataset, then we select one image belonging to the same class as positive sample and one belonging to a different class as negative sample. These images are selected randomly within these two sets.

The second strategy chooses a random image from the dataset, then extracts the most similar image of the same class and the most dissimilar from all other classes, excluding the class of the query image. To determine image similarity, we compute the Euclidean distance between them using feature vectors extracted from a CNN.

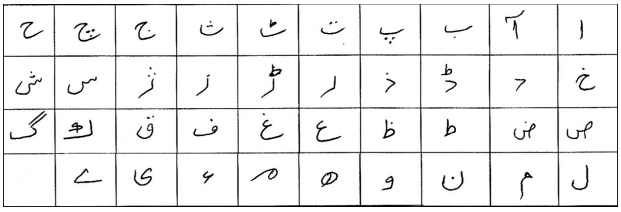


Figure 2: Typical handwritten response received from a student on a printed plain paper.

The third strategy differs from the second one on the selection of most similar image as positive image to the query image. This strategy selects the most dissimilar image in the same class as positive image. The vice-versa is for negative image. As reported in Equation 2, we expect to obtain best results with the third strategy because during the training process there would be higher error, making the backpropagation process more effective, while the second strategy would apply minimum adjustment within each step because of the similarity between query and positive image and the large difference between the query and the negative image.

IV. Dataset

The first dataset we use in experiments is the original MNIST [11] consisting of $60,000$ gray-scale images of handwritten digits $(0-9)$ and a corresponding set of $10,000$ test images with 28×28 pixels. MNIST dataset is extensively used in deep learning methods and is considered benchmark dataset. However, MNIST dataset has only few seemingly similar classes. This lead us to build a new handwritten dataset named Urdu-Characters, built in a similar way as MNIST dataset. The nature of characters in handwritten Urdu-Characters dataset is ideal for deep metric learning methods. There are some sets of classes available in Urdu-Character dataset which are seemingly similar however belong to different classes as shown in Figure 3.

Urdu-Characters dataset is collected on a printed plain paper with an $6in \times 2in$ box and 10×4 grid. To collect the data, a group of undergraduate students participated in the activity. In particular, students are asked to write Urdu characters in a specific sequence from right to left. We received 560 responses from students. The collected forms are then scanned at 300 dpi resolution in 8 bit gray scale image for further processing. Figure 2 shows a response example of a student having written all Urdu characters from right to left.

The vertical and horizontal projections of student responses are obtained to detect grid lines for character segmentation. Figure 4 and Figure 5 show these projections. Each projection is obtained summing all rows to first row and thus obtaining a plot. A similar procedure is used for horizontal projection. The projection lines with 85% or less sum were treated as separating lines and characters between them were separated. Extracted characters were normalized and converted into a 64×64 pixels image with

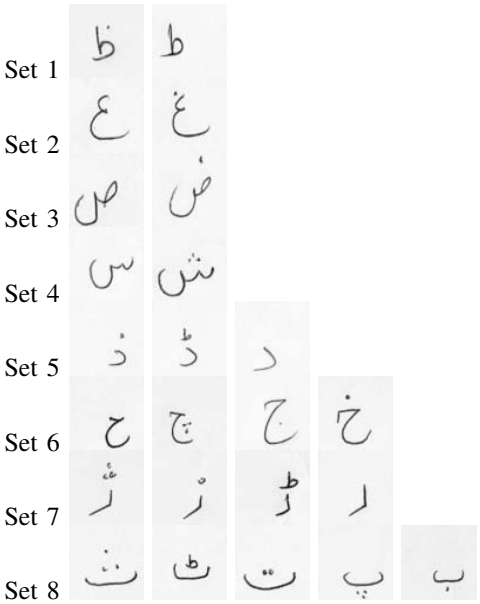


Figure 3: Interesting set of classes in Urdu-Characters dataset.

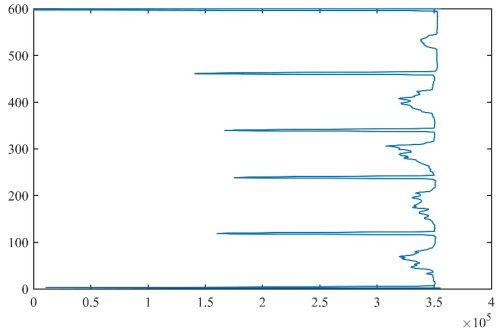


Figure 4: Vertical projection to detect vertical grid lines of handwritten response received from a student.

8 bit depth. Table 6 shows some examples of extracted handwritten characters. There are $20,324$ segmented characters grouped in 39 classes with $15,251$ characters for train and $5,073$ characters for test set.

V. Experiments

We compare the performance of a Siamese and Triplet network on MNIST and Urdu-Characters datasets. In addition, we want to compare the performance of Siamese and Triplet Networks built on top of different underlying CNNs architectures with different embedding dimensionality. To achieve these objectives, we performed a series of experiments on both datasets as follow:

- Build a Siamese network
- Compare the performance of Siamese network built on top of different underlying CNNs (Lenet and Alexnet)
- Build a Triplet network
- Compare the performance of a Triplet and Siamese network

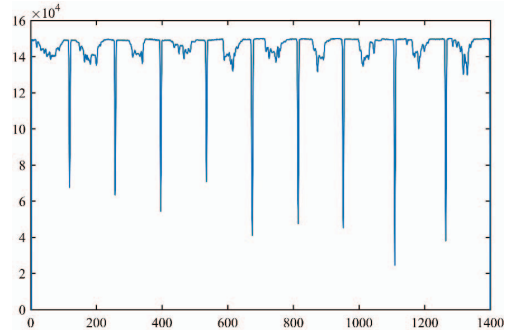


Figure 5: Horizontal projection to detect horizontal grid lines of handwritten response received from a student.

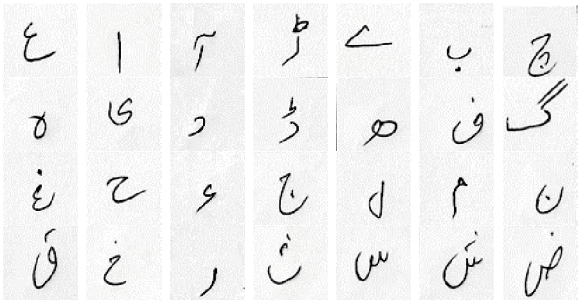


Figure 6: Extracted Urdu characters from a student response.

- Compare the performance of Triplet networks built on top of different underlying CNNs architectures

Table I: Siamese network settings built on top of Lenet for MNIST and Urdu-Characters datasets.

Dataset	Resolution	Network	Embedding	Pairs	Accuracy
MNIST	28×28	Lenet	256	$100,000$	96.23
Urdu-Character	64×64	Lenet	256	$100,000$	27.79

We use Caffe [16] and The NVIDIA Deep Learning GPU Training System deep learning frameworks, which contains efficient GPU implementations for training CNNs. In experiments, accuracy is employ to measure the performance of different network settings. Table I shows a Siamese network settings for MNIST and Urdu-Character datasets built on top of Lenet. We use $100,000$ pairs of similar and dissimilar randomly selected images to built a Siamese network. Output embedding dimensionality of the network is 256. The last column in Table I shows the accuracy obtained by Siamese network. The accuracy value of a Siamese network shows that network built on top of Lenet does not perform well on a complex dataset like Urdu-Characters. This leads us to built a Siamese network on top of more powerful network i.e. Alexnet for Urdu-Characters dataset. Table II shows that a Siamese network built on top of Alexnet produces better results compared to the same model built on top of Lenet.

Table II: Siamese network settings built on top of Lenet and Alexnet for Urdu-Characters.

Dataset	Resolution	Net.	Emb.	Triplet	Accuracy
Urdu-Character	64×64	Lenet	256	$100,000$	27.79
Urdu-Character	64×64	Alexnet	256	$100,000$	61.46

These results lead us to built a competitor of Siamese network i.e. a Triplet network. We use $100,000$ triplets consisting of query, positive and negative images to build a Triplet network. Table III shows a Triplet network settings for MNIST and Urdu-Character datasets built on top of Lenet. Accuracy values of both datasets for a Triplet Network is higher than accuracy values of a Siamese Network as shown in Figure 9. This proves that a Triplet Network is more powerful than a Siamese Network. We also built a Triplet network on top of Alexnet to obtain better results than a Triplet network built on top of Lenet. Table IV shows the accuracy values of a triplet network built on top of Lenet and Alexnet. This leads us to built a Triplet network on even more powerful network like Googlenet. However, to perform this experiment we need an image size of 256×256, hence, we up sample Urdu-Characters dataset images. Accuracy values in Table V show that a Triplet network built on top of Googlenet is more powerful than a network built on top of Alexnet.

Table III: Triplet network settings built on top of Lenet for MNIST and Urdu-Characters datasets.

Dataset	Resolution	Net.	Emb.	Triplet	Accuracy
MNIST	28×28	Lenet	256	$100,000$	98.23
Urdu-Characters	64×64	Lenet	256	$100,000$	53.45

Table IV: Triplet network settings built on top of Lenet and Alexnet for Urdu-Characters dataset.

Dataset	Resolution	Net.	Emb.	Triplet	Accuracy
Urdu-Characters	64×64	Lenet	256	$100,000$	53.45
Urdu-Characters	64×64	Alexnet	256	$100,000$	69.35

We compare the impact of the embedding dimensionality on Siamese and Triplet networks. This leads us to built a Triplet network on top of Alexnet and Googlenet and a Siamese network on top of Lenet and Alexnet with $128, 256, 512$ embedding dimensionality as shown in Figure 7 and 8. These results show that higher embedding dimensionality obtain better accuracy values. However, the choice of embedding dimensionality depends considerably on the application context. For example, using search by example system, a higher embedding dimensionality could make the process very slow.

Table V: Triplet network settings built on top of Alexnet and Googlenet for Urdu-Characters dataset.

Dataset	Resolution	Net.	Emb.	Triplet	Accuracy
Urdu-Characters	256×256	Alexnet	256	$100,000$	69.98
Urdu-Characters	256×256	Googlenet	256	$100,000$	77.06

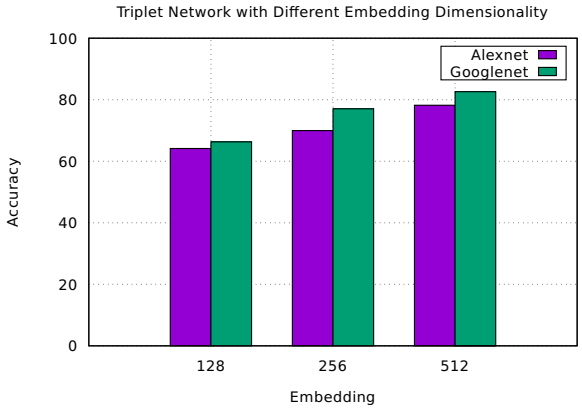


Figure 7: Triplet network built on top of Alexnet and Googlenet with $128, 256, 512$ embedding dimensionality for Urdu-Characters dataset. We employ the same network settings mentioned in Table V to built the network.

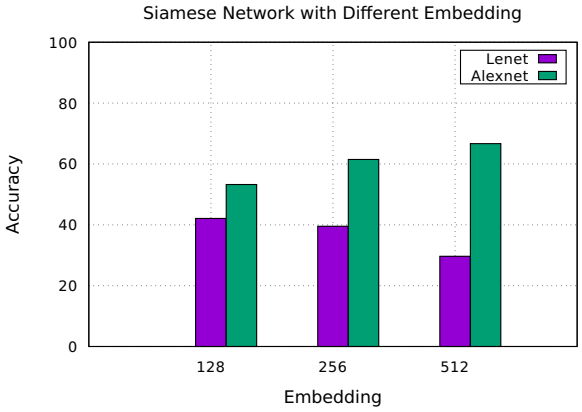


Figure 8: Siamese network built on top of Lenet and Alexnet with $128, 256, 512$ embedding dimensionalities for Urdu-Characters dataset. We employed same network settings mentioned in Table II to built the network.

Table VI: Comparison of Triplet sampling strategies. We built triplet network on top of Alexnet with $15,251$ triplets for training and $5,073$ triplets for test. Embedding dimensionality for these networks is 128.

Strategy	Accuracy
Strategy # 1	59.88
Strategy # 2	61.48
Strategy # 3	61.78

Finally, we compare the effect of different triplet sampling strategies on the performance of the triplet network. Table VI shows the performance of three sampling strategies discussed in section III-D. Results of these strategies are comparable however, strategy 3 is better than other two strategies because it violates the triplet constraints. However, we believe that the triplet selection strategy depends on the variation in the dataset. In our Urdu-Characters dataset, we have do not have high variability inside classes.

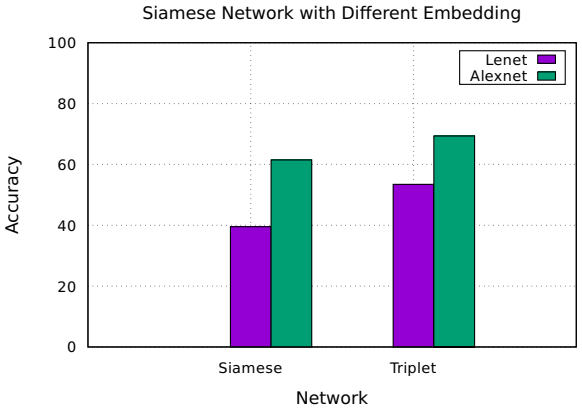


Figure 9: Comparison of Triplet and Siamese networks built on top of Lenet and Alexnet with 256 embedding dimensionality for Urdu-Characters dataset. We employed network settings mentioned in Table II and Table IV to built a Siamese and triplet network respectively.

VI. CONCLUSIONS

We built a handwritten Urdu-Characters dataset containing some sets of classes suitable for deep metric learning methods. We showed that Siamese network built Lenet performed well on MNIST dataset, but it did not reach good results on Urdu-Characters dataset however, a Siamese network built on top of Alexnet obtains significantly better results on Urdu-Characters. A similar phenomenon also happened for a Triplet network, where the difference between using different underlying CNN architectures such as Lenet, Alexnet or Googlenet is considerable in terms of overall accuracy. Furthermore, we compared three sampling strategies to create triplets to built a Triplet network, but we obtained comparable results. Usually the use of different sampling strategies lead to different accuracy values due to the variation in the dataset, however not in our case with Urdu-Character dataset due to the fact that it does not have high variability inside classes.

In a future release, an expanded Urdu-Characters dataset with $2,000$ student responses will be release. This will increase dataset size to $100,000$ instances.

REFERENCES

[1] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4004–4012.

[2] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, and S. Belongie, "Learning visual clothing style with heterogeneous dyadic co-occurrences," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4642–4650.

[3] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 31–35.

[4] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker, "Universal correspondence network," in *Advances in Neural Information Processing Systems*, 2016, pp. 2414–2422.

[5] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 539–546.

[6] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[7] A. Choromanska, A. Agarwal, and J. Langford, "Extreme multi class classification," in *NIPS Workshop: eXtreme Classification, submitted*, 2013.

[8] Y. Prabhu and M. Varma, "Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 263–272.

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition."

[12] B. Kulis *et al.*, "Metric learning: A survey," *Foundations and Trends® in Machine Learning*, vol. 5, no. 4, pp. 287–364, 2013.

[13] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a siamese time delay neural network," in *Advances in Neural Information Processing Systems*, 1994, pp. 737–744.

[14] S. Bell and K. Bala, "Learning visual similarity for product design with convolutional neural networks," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 98, 2015.

[15] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1386–1393.

[16] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.