Adversarial Training For Sketch Retrieval

Antonia Creswell & Anil Anthony Bharath

BICV Group, Bioengineering,
Imperial College London
ac2211@ic.ac.uk

Abstract. Generative Adversarial Networks (GAN) are able to learn excellent representations for unlabelled data which can be applied to image generation and scene classification. Representations learned by GANs have not yet been applied to retrieval. In this paper, we show that the representations learned by GANs can indeed be used for retrieval. We consider heritage documents that contain unlabelled Merchant Marks, sketch-like symbols that are similar to hieroglyphs. We introduce a novel GAN architecture with design features that make it suitable for sketch retrieval. The performance of this sketch-GAN is compared to a modified version of the original GAN architecture with respect to simple invariance properties. Experiments suggest that sketch-GANs learn representations that are suitable for retrieval and which also have increased stability to rotation, scale and translation compared to the standard GAN architecture.

Keywords: Deep learning, CNN, GAN, Generative Models, Sketches

1 Introduction

Recently, the UK's National Archives has collected over 70,000 heritage documents that originate between the 16th and 19th centuries. These documents make up a small part of the "Prize Papers", which are of gross historical importance, as they were used to establish legitimacy of ship captures at sea.

This collection of documents contain *Merchant Marks* (see Fig.4B), symbols used to uniquely identify the property of a merchant. For further historical research to be conducted, the organisation requires that the dataset be searchable by visual example (see Fig.1). These marks are sparse line drawings, which makes it challenging to search for visually similar Merchant Marks between documents. This dataset poses the following challenges to learning representations that are suitable for visual search:

1. Merchant marks are line drawings, absent of both texture and colour, which means that marks cannot be distinguished based on these properties.
2. Many machine learning techniques, and most notably convolutional neural networks (CNNs), require large amounts of labelled training data, containing on the order of millions of labelled images [7]. None of the Merchant Marks are labelled, and in many cases it is not clear what labels would be assigned to them. This motivates an unsupervised approach to learning features.

3. The marks are not segmented from the dataset, limiting the number of examples available, and making it difficult to train CNNs.
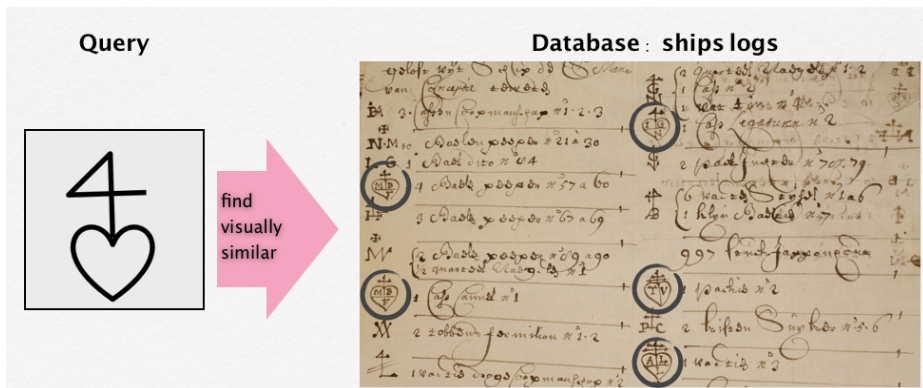

Fig. 1: An overview of the problem: the circled items contain examples of Merchant Marks; note that although some marks are distinct, they are still visually similar. We would like to retrieve visually similar examples, and find exact matches if they exist. Note that the two marks on the left are exact matches, while the others might be considered to be visually similar.

To perform visual search on the Merchant Marks, a representation for the marks that captures their structure must be learned. Previous work has demonstrated that deep convolutional neural networks (CNNs) are able to learn excellent hierarchical representations for data [15]. CNNs have proven useful for tasks such as classification [7], segmentation [9] and have been applied to retrieval of art work [2][1]. However, these methods rely on large amounts of labelled data for learning the weights. In the absence of sufficient labelled training data, we propose the use of unsupervised techniques with CNN architectures to learn representations for the Merchant Marks.

Unlike some previous approaches in which feature representations were learned by using labelled datasets that differ in appearance from the retrieval set, we used the Merchant Marks dataset itself to learn dataset-specific features. For example, Crowley et al. [2] trained a network similar to AlexNet [7] on examples from the photographic scenes of ILSVRC-2012 in order to learn features for the retrieval of art work; they also trained a network on photographs of faces to learn features for retrieving paintings of faces [1]. Yu et al. [14] suggested that features suitable for understanding natural images are not necessarily the most appropriate for understanding sketches.

Convolutional Auto-encoders (CAE) can be a useful tool for unsupervised learning of features. They are made up of two networks, an encoder which compresses the input to produce an encoding and a decoder, which reconstructs the input from that encoding. It has been shown [8] that shallow CAEs often learn
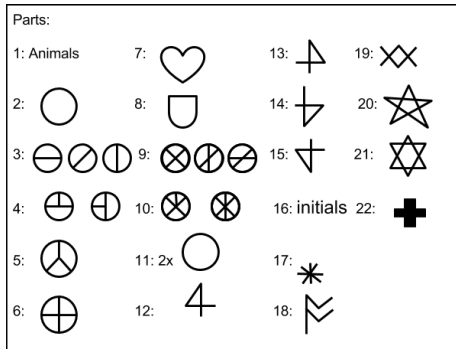
Fig. 2: Most marks in the Merchant Marks dataset are made of of the above sub-structures which we refer to as *parts*.

the delta function (a trivial solution) which is not a useful representation for the data. Instead, deep encoders are needed with strict regularisation on the activations. The Winner Take All CAE [8] imposes both spatial and life-time sparsity on the activations of the CAE in order to learn useful representations. Other regularisation techniques include the Variational Auto-encoder [6], which imposes a prior distribution on the encoding.

An alternative method, which learns representations from data without the need for regularisation, is the Generative Adversarial Network [3] (GAN). Deep convolutional generative adversarial networks [10] have been shown to learn good representations for data. In this paper, we propose the use of GANs for learning a representation of the Merchant Marks that can be used for visual search.

The key contribution is to show that GANs can be used to learn a representation suitable for visual search. We apply this novel idea to the Merchant Mark dataset, and compare two GAN architectures. The first GAN is designed to learn a representation for sketches, based on reported architectural considerations specific to sketches [14]. The second GAN is a modified version of the network proposed by Radford et. al [10] often used for learning representations for natural images. The representations are evaluated by comparing their invariance to shift, scale and rotation as well as the top 8 retrieval results for 15 examples.

2   Generative Adversarial Networks

Generative Adversarial Networks (see Fig. 3), (GANs) where first introduced by Goodfellow et al [3], as a generative model that learned an excellent representation for the training dataset. GANs consist of two networks, a generative network, G and a discriminative network, D. The goal of the generative network is to learn the distribution of the training data, $p_{data}(x)$ where $x \in R^{dx}$ and dx is the dimensions of a data sample.
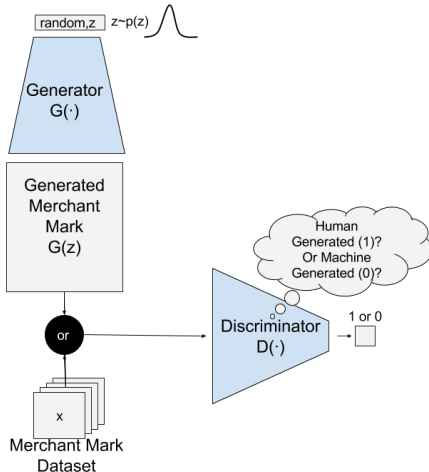
Fig. 3: Generative Adversarial Network: A random sample z is drawn from a prior distribution and fed into the generator, G to generate a sample. The discriminator will take a sample either from the generator, $G(z)$, or from the Merchant Mark dataset, $p_{data}(x)$, and predict whether the sample is machine or human generated. The discriminator's objective is to make the correct prediction, while the generator's objective is to generate examples that fool the discriminator.

In a GAN, the generator takes as input a vector, $z \in R^{dz}$ of dz random values drawn from a prior distribution $p_z(z)$, and maps this to the data space, $G : R^{dz} \rightarrow R^{dx}$. The discriminator takes examples from both the generator and real examples of training data and predicts whether the examples are human (or real) (1) or machine generated (0), $D : R^{dx} \rightarrow [0, 1]$.

The objective of the discriminator is to correctly classify examples as human or machine generated, while the objective of the generator is to fool the discriminator into making incorrect predictions. This can be summarised by the following value function that the generator aims to minimise while the discriminator aims to maximise:

$$\min_{G} \max_{D} \mathbb{E}_{x \sim p_{data}(x)} \log D(x) + \mathbb{E}_{z \sim p_z(z)} \log(1 - D(G(z)))$$

Training an adversarial network, both the generator and the discriminator learn a representation for the real data. The approach considered here will use the representation learned by the discriminator.

3   Methods

Here, we show how the discriminator, taken from a trained GAN, can be modified to be used as an encoder for sketch retrieval. An overview of the methods used can be seen in Fig.4.
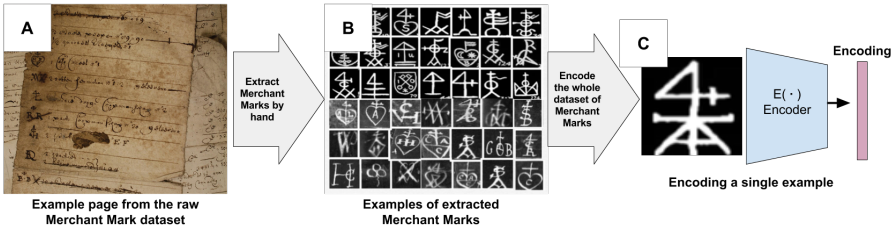
Fig. 4: Overview: (A) Shows examples of raw Merchant Mark data, photographs of the documents. (B) Shows examples extracted by hand from the raw Merchant Mark dataset, a total of 2000 examples are collected. (C) An encoder is simply a discriminator, taken from a trained GAN with the final layer removed. Representations for both query and data samples are obtained by passing examples through the encoder, the representations are used for retrieval.

3.1   Dataset Acquisition

The raw Merchant Mark dataset that we have been working with consists of 76 photographs of pages from the raw Merchant Mark dataset, similar to the one shown in Fig.4A, which contain multiple Merchant Marks at different spatial locations on the page. The focus of this paper is on retrieval of visually similar examples rather than localisation, so the first step involved defining box regions from which Merchant Mark training examples could be extracted. The extracted examples are re-size to be 64×64 pixels to form a suitable dataset for training a GAN. In total there are 2000 training examples (see Fig.4B).

3.2   Learning an Encoder

Training A GAN To learn an encoding, the generator and the discriminator of a GAN are trained iteratively, as proposed by Goodfellow et al. [3]. See pseudo-code in Alg.1.

Network Architecture Both the generator, and the discriminator are convolutional neural networks [13], using convolutions applied with strides rather than pooling as suggested by Radford et al. [10]. In the discriminator, the image is mapped to a single scalar label, so the stride applied in the convolutional layer of the discriminator must be grater than 1. A stride of 2 is used in all convolutional layers of the discriminator. In the generator, a vector is mapped to an image, so a (positive) step size less than 1 is needed to increase the size of the image after each convolution. A stride of 0.5 is used in all convolutional layers of the generator.

Encoding Samples Having trained both the generator and the discriminator, the discriminator can be detached from the GAN. To encode a sample, it is

for *Number of training iterations* **do**

    for *k iterations* **do**

        sample $p_z(z)$ to get m random samples $\{z_1...z_m\}$

        sample $p_{data}(x)$ to get m random samples $\{x_1...x_m\}$

        calculate the discriminator error:

$$J_D = -\frac{1}{2m}\left(\sum_{i=1}^{m}\log D(x_i) + \sum_{i=1}^{m}\log(1 - D(G(z_i)))\right)$$

        update θ_D using Adam [5] update rule.

    end

    sample $p_z(z)$ to get m random samples $\{z_1...z_m\}$

    calculate the generator error:

$$J_G = -\frac{1}{m}\sum_{i=1}^{m}\log(D(G(z_i)))$$

    update θ_G using Adam [5] update rule.

end

Algorithm 1: Training a GAN: After Goodfellow et al. [3] with changes to the optimisation, using Adam [5] instead of batch gradient descent. Note, m is the batch size and θ_G, θ_D are the weights of the generator, G and discriminator, D.

passed through all but the last layer of the discriminator. The discriminative network without the final layer is called the *encoder*. Both the query examples and all examples in the dataset can be encoded using the this encoder. The encoding is normalised to have unit length by dividing by the square root of the sum of squared values in the encoding.

3.3 Retrieval

The objective is to retrieve samples that are visually similar to a query example. To retrieve examples similar to the query, similarity measures are calculated between the representation for the query and representations for all samples in the dataset. Examples with the highest similarity scores are retrieved. The focus of this paper is on learning a good representation for the data, for this reason a simple similarity measure is used, the (normalised) dot product.

4 Experiments And Results

The purpose of these experiments is to show that GANs can be used to learn a representation for our Merchant Mark dataset from only 2000 examples, that can be used to precisely retrieve visually similar marks, given a query. We compare invariance of feature representations learned and retrieval results from two different networks to show that there is some benefit to using a network designed specifically for learning representations for sketches.

4.1 GAN Architectures

Two different architectures were compared:

sketch-GAN We propose a novel GAN architecture inspired by Sketch-A-Net [14], a network achieving state of the art recognition on sketches. Sketch-A-Net employs larger filters in the shallower layers of the discriminative network to capture structure of sketches rather than fine details which are absent in sketches. This motivated our network design, using larger filters in the lower levels of the discriminator and the deeper levels of the generator. This network will be referred to as the *sketch-GAN*. This network has only 33k parameters.

thin-GAN A network similar to that proposed by Radford et al. [10] is used. This network has very small filters, consistent with most of the state-of-the-art natural image recognition networks [12]. The original network has 12.4M parameters which would not compare fairly with the *sketch-GAN*, instead a network with 1/16th of the filters in each layer is used, this will be referred to as the *thin-GAN* and has 50k parameters. Full details of the architecture are given in Table.1.

4.2 Details of Training

In adversarial training the generator and discriminator networks are competing against eachother in a mini-max game, where the optimal solution is a Nash Equilibrium [11]. Adversarial networks are trained iteratively alternating between the generator and discriminator using gradient descent which aims to minimise the individual cost functions of the generator and discriminator, rather than finding a Nash Equilibrium [11]. For this reason convergence, during adversarial training cannot be guaranteed [11][4]. During training we found that networks did not converge, for this reason networks were trained for a fixed number of iterations, rather than till the networks converged. The networks are trained for 2000 iterations with batch size of 128 according to Alg. 1 [3], with $k = 1$, $dz = 2$, learning rate $= 0.002$, and $p_z(z) \sim U(0, 1)$. The networks were still able to learn features useful for retrieval despite not converging.

4.3 Feature Invariance

Merchant Marks are hand drawn, which means that the marks are likely to vary in both scale and orientation. It is therefore important to consider the rotation and scale invariance of the representations that result from training. When searching a document for Marks, one approach may be to apply a sliding box search. The step size in sliding the search box will affect the computational feasibility of the search. If a representation used for search is invariant to larger shifts, then a sliding box search can be performed with a larger step size, making the search more efficient. For this reason, shift invariance of the two representations is also compared.

Table 1: A summary of the network architectures used in this study. fc=fully connected layer, c=convolutional layer with stride 2, d=convolutional layer with stride 0.5, unless stated otherwise; for all cases, dz, the dimension of the random valued vector input to the generator is 2. The ReLU activation function is used in all hidden layers of all networks and the sigmoid activation is used in final layer of each network.

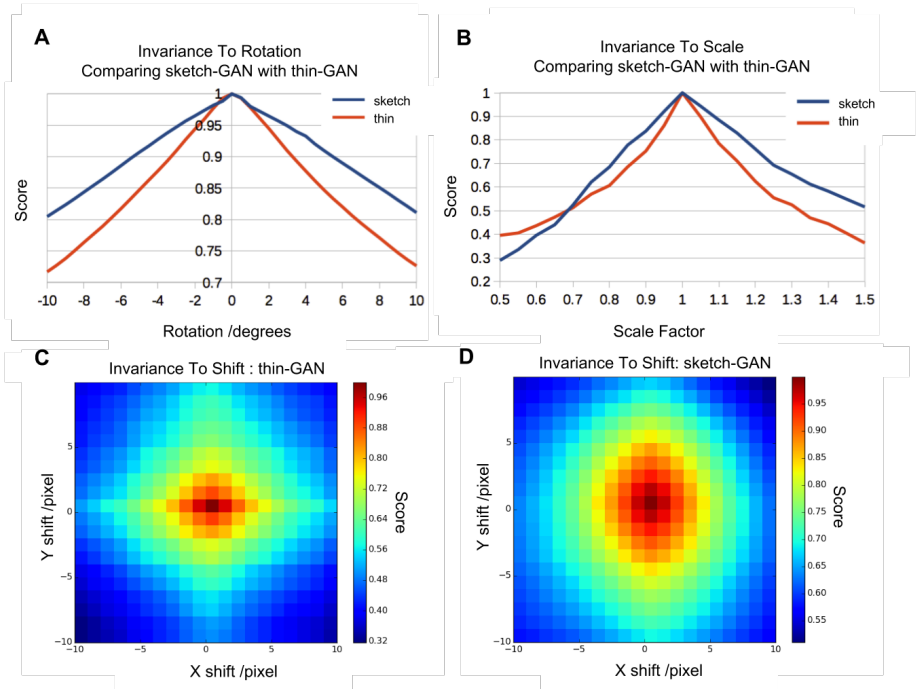| thin-GAN:G	thin-GAN:D
fc: $1024 \times dz$, reshape(64,4,4)	c: $8 \times 1 \times 3 \times 3$
d: $32 \times 64 \times 3 \times 3$	c: $16 \times 8 \times 3 \times 3$
batch normalisation	batch normalisation
d: $16 \times 32 \times 3 \times 3$	c: $32 \times 16 \times 3 \times 3$
batch normalisation	batch normalisation
d: $8 \times 16 \times 3 \times 3$	c: $64 \times 32 \times 3 \times 3$
batch normalisation	batch normalisation, reshape(1024)
d: $1 \times 8 \times 3 \times 3$	fc: 1×1024
sketch-GAN:G	sketch-GAN:D
fc: $128 \times dz$, reshape(8,4,4)	c: $8 \times 1 \times 9 \times 9$ (stride=1)
d: $16 \times 8 \times 3 \times 3$	c: $16 \times 8 \times 5 \times 5$
batch normalisation	batch normalisation
d: $16 \times 16 \times 5 \times 5$	c: $16 \times 16 \times 5 \times 5$
batch normalisation	batch normalisation
d: $16 \times 16 \times 5 \times 5$	c: $16 \times 16 \times 5 \times 5$
batch normalisation	batch normalisation, reshape(1,1024)
d: $16 \times 16 \times 5 \times 5$	fc: 1×1024
batch normalisation	-
d: $1 \times 16 \times 9 \times 9$ (stride=1)	-

Fig. 5: Invariance: Shows invariance of the *sketch-GAN* and *thin-GAN* representations to A) rotation, B) scale and C,D) translation.

Invariance To Rotation To assess the degree of rotation invariance within the two representations, 100 samples were randomly taken from the Merchant Mark dataset and rotated between the angles of -10 and 10 degrees. At each 0.5 degree increment, the samples were encoded and the similarity score between the rotated sample and the sample at 0 degrees was calculated. The similarity score used was the normalised dot product, since this was also the measure used for retrieval. The results are shown in the top left of Fig.5. It is clear that the *sketch-GAN* encoding is more tolerant to rotation than the *thin-GAN* encoding. Note that the background of the rotated samples were set to 0 to match the background of the samples.

Invariance To Scale A similar approach was used to assess the degree of scale invariance within the two networks. Again, 100 samples were randomly taken from the Merchant Mark dataset, and scaled by a factor between 0.5 and 1.5. At each increment of 0.05, the scaled sample was encoded and a similarity score was calculated between the scaled samples and the sample at scale 1. The results are shown in the top right of Fig.5. Note, that when the scaling factor is < 1 the scaled image is padded with zeros to preserve the 64×64 image size. When scaling with a factor > 1, the example is scaled and cropped to be of size 64×64. The bounding box of the unscaled marks is tight, which means that at higher scaling factors parts of the marks are sometimes cropped out. Despite this, the *sketch-GAN* encoding is able to cope better with up-scaling compared to down-scaling. The *sketch-GAN* encoder generally outperforms the *thin-GAN* encoder, particularly for up-scaling.

Invariance To Shift Finally, we compared the shift invariance of the two encoders. Sampling 100 marks from the merchant mark dataset, and applying shifts between -10 and 10 pixels in increments of 1 pixel in both the x and y directions. The results are shown as a heat map in Fig.5, where the *sketch-GAN* encoding appears to be more invariant to shift than the *thin-GAN* encoding.

4.4   Retrieval

For the retrieval experiments, 500 queries were taken at random from the training dataset and used to query the whole dataset using features from the *sketch-GAN*. The top 9 matches were retrieved, where the first retrieval is the example itself and the rest are examples that the system thinks are similar. The results from some of these queries are shown in Fig.6b. The same query examples were used to query the dataset using the features from the *thin-GAN*, the results of these queries are shown in Fig.6a.

Retrieval results using trained sketch-GAN encoder Results show that using the *sketch-GAN* encoder for Merchant Marks retrieval (Fig.6b) allows retrieval of examples that have multiple similar parts for example results for

(a) thin-GAN   (b) sketch-GAN
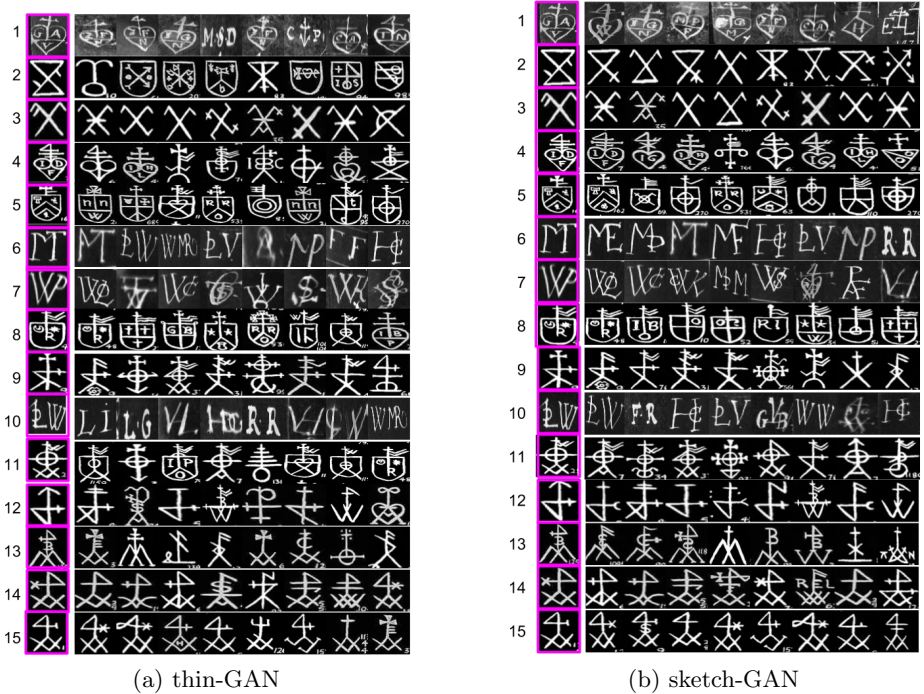
Fig. 6: Retrieval examples using different GAN architectures. Each sub-figure shows 15 retrievals where the 1st example, in a purple box, in each row is the query and the following images on each row are the top 8 retrievals. (a) Shows retrievals using the thin-GAN encoder and (b) shows retrievals using the sketch-GAN encoder.

queries #4,#8,#11,#14 and #15 consistently retrieve examples with at least two similar parts (Fig.2). Specifically, most retrievals for query #15, Fig.6b have parts 12 and 19 from Fig.2. Exact matches are found for retrievals #4,#5,#8 and #10. Specifically, query #10 finds an exact match despite the most similar example being shifted upwards and rotated slightly. Retrievals for query #6 finds an exact match but does not rank the retrieval as high as non-exact matches, suggesting that there is still room for improvement in the representations that are learned.

Retrieval results using trained thin-GAN encoder On visual inspection of the retrieval results that use the *thin-GAN* encoder, it is clear that they under perform compared to the *sketch-GAN* for the same query examples, with fewer visually similar examples. The *thin-GAN* encoder fails to find exact matches for 4,5 and 10. Failure to find a match for 10 further suggests that the *thin-GAN* is less invariant to rotation.

5   Conclusions

Convolutional networks contain, at a minimum, tens of thousands of weights. Training such networks has typically relied on the availability of large quantities of labelled data. Learning network weights that provide good image representations in the absence of class labels is an attractive proposition for many problems. One approach to training in the absence of class labels is to encourage networks to compete in coupled tasks of image synthesis and discrimination. The question is whether such Generative Adversarial Networks can learn feature representations suitable for retrieval in a way that matches human perception.

We have found that GANs can indeed be used to learn representations that are suitable for image retrieval. To demonstrate this, we compared the representation learned by GANs that were trained on Merchant Marks. We compared two related architectures, *sketch-GAN* and *thin-GAN*; *sketch-GAN* has an architectural design that is more appropriate for performing generation and discrimination of sketches. Our experiments showed that GANs are suitable for retrieval of both visually similar and exact examples. Experiments also showed that the features that were learned by the *sketch-GAN* were, on average, more robust to small image perturbations in scale, rotation and shift than the *thin-GAN*. Further, retrieval results when using the *sketch-GAN* appeared more consistent than in using *thin-GAN*.

More generally, the experiments suggest that adversarial training can be used to train convolutional networks for the purpose of learning good representations for the retrieval of perceptually similar samples; this can be achieved without the level of labelling and examples required for non-adversarial training approaches. This broadens the scope of deep networks to problems of perceptually similar retrieval in the absence of class labels, a problem that is increasingly of interest in heritage collections of images.

References

1. Crowley, E.J., Parkhi, O.M., Zisserman, A.: Face painting: querying art with photos. In: British Machine Vision Conference (2015)
2. Crowley, E.J., Zisserman, A.: In search of art. In: Workshop on Computer Vision for Art Analysis, ECCV (2014)
3. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems. pp. 2672–2680 (2014)

4. Goodfellow, I.J.: On distinguishability criteria for estimating generative models. arXiv preprint arXiv:1412.6515 (2014)

5. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

6. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)

7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)

8. Makhzani, A., Frey, B.J.: Winner-take-all autoencoders. In: Advances in Neural Information Processing Systems. pp. 2773–2781 (2015)

9. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1520–1528 (2015)

10. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)

11. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. arXiv preprint arXiv:1606.03498 (2016)

12. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

13. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806 (2014)

14. Yu, Q., Yang, Y., Song, Y.Z., Xiang, T., Hospedales, T.M.: Sketch-a-net that beats humans. In: Proceedings of the British Machine Vision Conference (BMVC). pp. 7–1 (2015)

15. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Computer vision–ECCV 2014, pp. 818–833. Springer (2014)