Machine Vision and Applications (2020) 31:5
https://doi.org/10.1007/s00138-019-01053-5

ORIGINAL PAPER

Convolutional networks for appearance-based recommendation and visualisation of mascara products

Christopher J. Holder[1] · Stephen Ricketts[2] · Boguslaw Obara[1]

Received: 31 May 2019 / Revised: 15 October 2019 / Accepted: 3 December 2019 / Published online: 21 January 2020

Abstract

In this work, we explore the problems of recommending and visualising makeup products based on images of customers. Focusing on mascara, we propose a two-stage approach that first recommends products to a new customer based on the preferences of other customers with similar visual appearance and then visualises how the recommended products might look on the customer. For the initial product recommendation, we train a Siamese convolutional neural network, using our own dataset of cropped eye regions from images of 91 female subjects, such that it learns to output feature vectors that place images of the same subject close together in high-dimensional space. We evaluate the trained network based on its ability to correctly identify existing subjects from unseen images, and then assess its capability to identify visually similar subjects when an image of a new subject is used as input. For product visualisation, we train per-product generative adversarial networks to map the appearance of a specific product onto an image of a customer with no makeup. We train models to generate images of two mascara formulations and assess their capability to generate realistic mascara lashes while changing as little as possible within non-lash image regions and simulating the different effects of the two products used.

Keywords Deep learning · Generative adversarial networks · Siamese networks · Recommender systems · Cosmetics

1 Introduction

In this paper, we describe, discuss and evaluate our work towards a cosmetic customer advisor system intended for use in retail environments. Focusing specifically on mascara products, our approach comprises two processes: product recommendation and product visualisation. Our recommendation approach uses a Siamese network [2] to identify prior customers with similar visual features to the current customer so that products preferred by those prior customers can be recommended. Our visualisation approach uses a generative adversarial network (GAN) [3] to generate realistic images demonstrating how a customer might look with a specific product applied. We envision such a system being deployed within an in-store photo booth to constrain as much as possible lighting, camera angle and other variables; however, future work may explore how such a system could be made robust to the variation inherent in a solution deployed on customer mobile phones.

Our models are trained and evaluated using our own dataset comprising multiple colour facial images of 91 Caucasian women with and without mascara applied, from which we automatically extract the left and right eye regions using a histogram of oriented gradients (HOG) and support vector machine (SVM)-based eye detector. The resultant cropped eye images are used to train our Siamese and GAN models, with a subset held back for testing.

Figure 1 shows some example images from our dataset (top row) and the subsequent mascara images generated by our GAN (bottom row).

This work is an extended version of our paper Visual Siamese Clustering for Cosmetic Product Recommendation [1] presented at the 1st International Workshop on Advanced Machine Vision. This paper expands on the discussion of the proposed approach for mascara recommendation and details additional related work that uses generative adversarial networks to visualise recommended products.

✉ Boguslaw Obara
boguslaw.obara@durham.ac.uk

Christopher J. Holder
c.j.holder@durham.ac.uk

[1] Department of Computer Science, Durham University, Durham, UK

[2] Walgreens Boots Alliance, Nottingham, UK

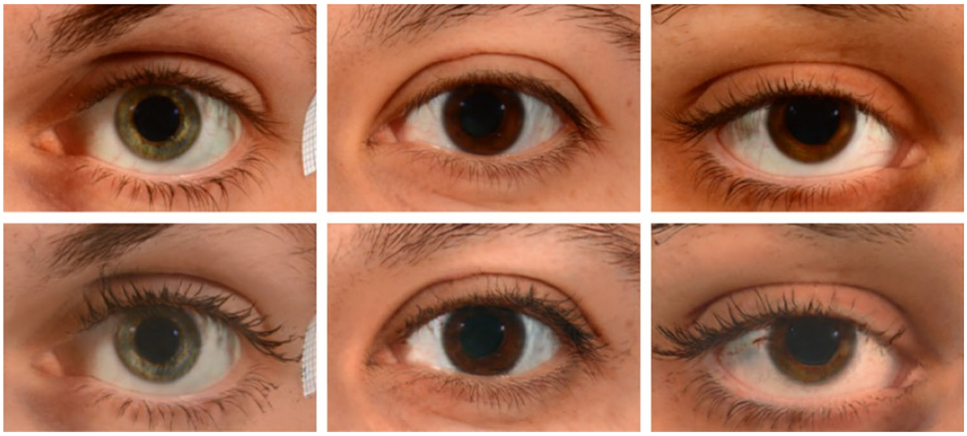

Fig. 1 Example eye images from our dataset (top) along with corresponding images generated by our GAN showing how they might look with mascara applied

2 Prior work

2.1 Recommendation

Recommender systems that make use of large amounts of data about a person to predict their preferences have been an effective method for businesses to increase customer engagement and retention since they were first proposed in the 1990s [4]. With the advent of big data and deep learning, sophisticated models capable of highly accurate predictions have been deployed within numerous industries [5, 6]; however, these approaches still require large numbers of datapoints for each customer, which are often not available to traditional bricks-and-mortar retail businesses.

For certain product types, however, such as cosmetics or clothing, visual information, such as that captured by in-store cameras or a customer's mobile device, may be all that is needed to make accurate product recommendations. Colour Match [7] is a mobile cosmetic advisory system that uses a colour-based statistical analysis of a calibrated mobile photograph to recommend a foundation product. McAuley et al. [8] propose an approach to matching clothing and accessories based on human notions of complementarity by computing an embedding that places the output vectors of a pretrained convolutional neural network (CNN) for two products close together if online shopping data demonstrates a correlation between interests in the two products.

In the broader area of visual matching, the concept of the Siamese network was first proposed by Bromley et al. [2], applied to the problem of signature verification, with the feature vector generated from a signature being compared with the stored feature vector corresponding to a given signer to determine whether the signature is genuine or not. Siamese, along with related triplet network [9], based approaches have since been proposed for solving a broad array of problems including facial recognition [10], unsupervised learning [11], one-shot image recognition [12] and image retrieval [13].

2.2 Visualisation

In addition to recommending a product, we aim to accurately visualise the application of that product on the customer's eyelashes using a GAN. First proposed by Goodfellow et al. [3] in 2014, a GAN in general comprises a deep generator network, which aims to learn the distribution of a target dataset such that it can generate plausible samples, and a deep discriminator network, which aims to learn discriminant features between the distributions of genuine and generated samples. GAN-based approaches have demonstrated promising results at tasks including anomaly detection [14], data augmentation [15], super resolution [16] and image domain translation [17].

Visualisation of makeup products as they would appear on a real customer's face is an active area of research within the cosmetics industry. One example of a GAN being applied in this area is BeautyGAN [18], which generates an image combining the face shown in one image with the makeup style shown in a second image. Liu et al. [19] demonstrate a different approach to the same problem, by parsing facial features from both images using a deep segmentation network and transferring extracted makeup features from one image to the other, employing a global smoothness constraint to maintain plausibility. In both cases, an overall 'makeup style' is transferred, and so neither lends itself to the problem of visualising individual makeup products. Other work in this area [20] uses an augmented reality-based approach which superimposes cosmetic products onto a live video of a user's face; however, these can often lead to an exaggerated and unrealistic appearance.

3 Dataset

We have compiled a dataset comprising frontal facial RGB colour images of 91 Caucasian women between the ages of 18 and 60, along with quantitative and qualitative data about their experiences of different mascara products. Each subject was photographed before and after application of one of two mascara formulations on four consecutive days, giving us eight images in total per subject.

We use only the image data in training our models, while the additional product preference data would be used to make recommendations once a customer has been matched to their most similar subjects.

From each image we generate 2 crops, 1 for each eye, giving us 16 eye images per subject; 8 with and 8 without mascara. Training of our Siamese network is conducted with 7 of these no-mascara images, while the 8th is kept for validation. We train 2 GAN models to generate images corresponding to each of the 2 mascara formulations used in the study. Each is trained using all eight non-mascara images per subject along with the four mascara images where the corresponding formulation has been applied.

We have a second set of images from a similar study, comprising 1 no-mascara image of each of 99 subjects not included in the first set, which we use for qualitative evaluation.

4 Method

In this section, we describe our approach that aims to recommend and visualise mascara products based on a single facial image of a potential customer. From this image, we use a sliding window HOG/SVM eye region detector to create a cropped image around each eye.

The recommender part of our system uses a Siamese network to place an image of the customer's eye within a learned embedding that aims to place visually similar eyes near to each other. The closest prior customers within this embedding can be retrieved allowing for recommendations to be made for products that they preferred.

The visualiser part of our system uses a collection of GANs, each trained to generate plausible images of eyelashes with a specific mascara formulation applied. An image of the customer's eye with no make-up is used as input, and the generated image can be displayed to the customer to give an idea of how they would look if they applied a particular product. In this work, we demonstrate our approach using 2 distinct mascara formulations.

4.1 Eye detector

The initial input is a set of images of whole faces; however, we are only interested in the region surrounding each of the eyes. Manually cropping every image in our dataset would consume a lot of time and may not be viable in a deployed system, so we use a histogram of oriented gradients (HOG) [21] feature descriptor and support vector machine (SVM) [22] classifier to identify the two regions of interest in each image for cropping.

We manually locate the eyes in a small subset (~ 10%) of our training data, which we subsequently crop and scale to dimensions of 48 × 32 pixels from which HOG features are computed and used to train an SVM classifier. Each HOG descriptor comprises a 540-dimensional vector encoding a 9-bin histogram describing the gradient features present in an 8 × 8 pixel cell computed within a 16 × 16 pixel sliding window with a stride of 8 pixels.

This vector is used as input to a one-class linear SVM, which during training attempts to determine the optimum boundary between observed samples and the rest of the feature space.

Images from which we want to extract the eyes are resized to 128 × 128 pixels, and HOG features are computed in a sliding window of 48 × 32 pixels with a stride of 1. We assume that each input image is closely cropped to a single face that this face contains two eyes which are open. The HOG descriptor computed at each position is used as input to our trained SVM which outputs the signed distance between a sample and the learned decision boundary, which we interpret as a confidence value that the given window is centred on an eye.

These values are used to create a heatmap H where each pixel value is the sum of confidence scores for all windows that contain it, as shown in Eq. (1): For every sliding window position R_i a confidence value C_i is computed, and the value of a heatmap pixel $H_{x,y}$ is the sum of all those C_i for which the corresponding R_i includes the point (x,y).

$$H_{x,y} = \sum_{i=0}^{n} \begin{cases} C_i & \text{if } (x, y) \in R_i \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

We again apply a sliding window of the same dimensions to identify the highest scoring region in the heatmap. We sum the pixel values at each window location, and the two highest scoring non-overlapping locations are taken as the subject's eyes, as illustrated in Fig. 2.

We use our approach to crop the eyes from the remainder of our dataset, manually checking the output cropped images for accuracy and determining a crop to be successful if an eye is fully contained within it. Successful crops were output for 97% of the unseen images in our dataset, leaving 3% that required manual cropping. Failure cases included images of

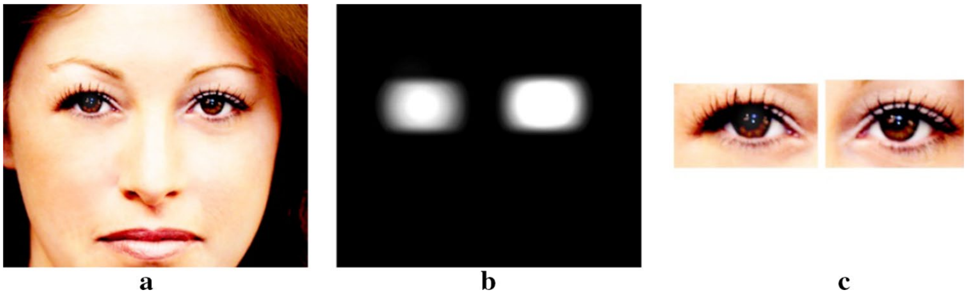

Fig. 2 Our eye detector uses HOG features and an SVM classifier to detect and crop eye regions from an image of the whole face (**a**). A heatmap **b** is generated from the confidence values output by a trained SVM, and the regions with the highest response are cropped **c** to give two eye images

subjects with prominent eyebrows, which would appear to generate similar HOG features to eyes, and images where a subject's eyes were closed, which we subsequently removed from the dataset.

Some example failure cases are shown in Fig. 3, demonstrating a closed eye, (although in this case the detector found the correct region it still had to be removed from the dataset), a prominent eyebrow that appears to have confused the detector, and an image where the subject's nose was mistaken for an eye.

4.2 Recommendation

Our approach for recommending products is based on a Siamese network [2] that has learned to place images of eyes within an embedding such that visually similar eyes are closer to each other than visually dissimilar eyes. When an image of a new customer is placed within this embedding, visually similar prior customers can be retrieved and products they are known to have preferred can be recommended to the new customer.

It has been shown [23–25] that the mere act of recommending a product to an undecided consumer can have a significant impact on the likelihood of a purchase taking place, and when there is a convincing 'story' behind a recommendation this effect is increased. As such, the aim of this work is not to necessarily recommend the single product that a customer will have the best experience with—especially when considering that mascara preferences can be influenced by many subjective factors that cannot be inferred from a single photograph—but to demonstrate an ability to match customer images to others that are visually similar as part of the convincing 'story' that can back up a product recommendation and influence a purchasing decision.

4.2.1 Siamese network

First proposed in [2], a Siamese network comprises two identical neural networks which share weights and are optimised such that when a pair of input samples belong to the same category, the output is similar by some distance metric, and when an input pair do not belong to the same category the output vectors are dissimilar. In our case, the network is optimised such that output vectors are similar for two images of the same person's eyes (either one image of each eye or two separate images of the same eye) and dissimilar for images of different people's eyes, with similarity being determined by Euclidean distance within an embedding space.

Our reasoning for using this method is that multiple images of eyes belonging to the same person should have strong visual similarity, and so by learning to place such images close together in the embedding space, our model should implicitly learn an embedding that places visually similar eyes close together.

The network model we use is VGG16 [26], first proposed by Simonyan et al. in 2014 for the purpose of large-scale image recognition, which is shown in a Siamese configuration in Fig. 4. The network comprises thirteen 3×3 convolutions interspersed with five 2×2 max-pooling layers, followed by three fully connected layers, the last of which outputs the final n-dimensional output vector. Batch normalisation is used between convolutions, rectified linear units provide nonlinearity and dropout is used during training to reduce the likelihood of overfitting. We use RGB input images with dimensions of 384×256 pixels, and evaluate models of output dimensionality n between 8 and 64.

During a training iteration using input image X_1, it is decided at random with $p = 0.5$ whether the second input

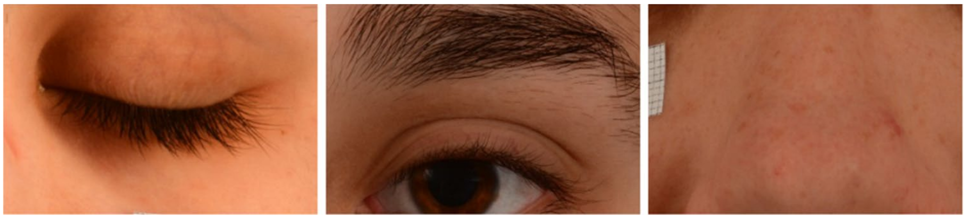

Fig. 3 Examples of failure cases of our HOG-based eye region detector

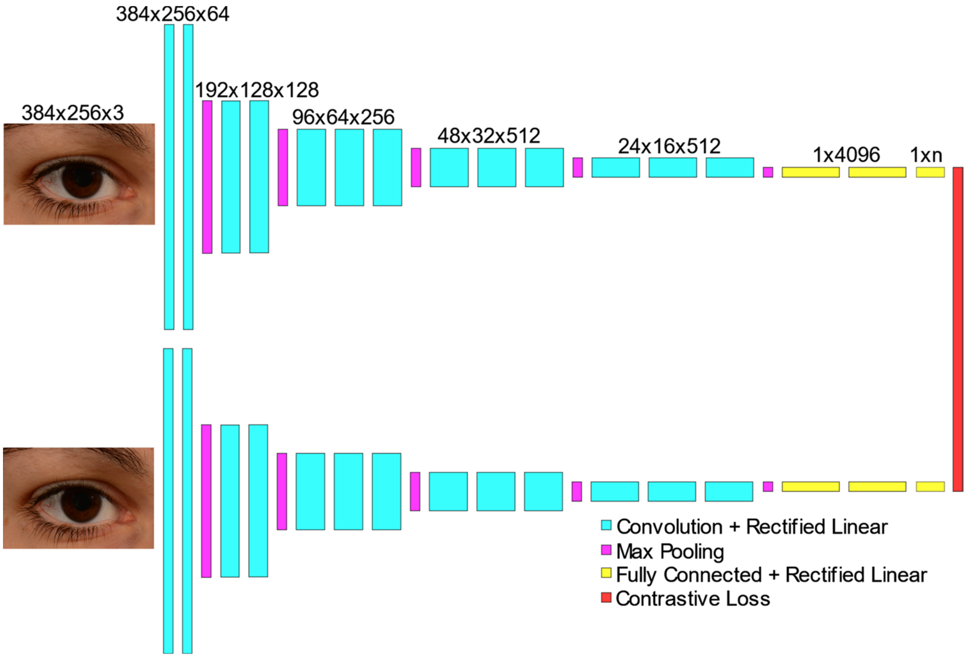

Fig. 4 The configuration of our Siamese network during training

image X_2 will be selected at random from the set of different images of the same subject or selected at random from the set of images that contain different subjects. To introduce greater variety into the data, we also mirror each input image at random with $p = 0.5$. X_1 and X_2 are each used as input for one forward pass of the network, generating two n-dimensional vectors, Y_1 and Y_2. We compute the Euclidean distance, d, between Y_1 and Y_2 and use a contrastive loss function similar to that of [28] (Eq. 2) which gives us loss l with respect to target t, where if X_1 and X_2 are images of the same subject $t = 0$, otherwise $t = 1$. A margin $m = 2$ is used such that the loss is 0 in cases where $t = 1$ and d is greater than m, as in cases where both samples come from different subjects we are only interested in ensuring that the distance between Y_1 and Y_2 is greater than or equal to m.

$$l = (1 - t) * d^2 + t * \max\{0, m - d\}^2 \tag{2}$$

We use backpropagation to optimise network parameters using the Adam optimiser [29] with weight decay to regularise parameters and learning rate decay to improve network convergence as training progresses. We initialise our network with random weights and train for 1000 epochs (an epoch being a period of training during which every training sample has been used as X_1 input once, although this may not be the case for X_2 due to it being selected at random), and save the model state at every 100 epochs so that we can evaluate network performance throughout the training process.

4.2.2 Clustering

Once our model is trained, we perform one forward pass for each of our training images, along with mirrored versions, and record the output vector, from which we compute the mean and standard deviation in each dimension for each subject. In this case, we are creating clusters for the 91 subjects whose data were used to train the model; however, clusters could be added for new subjects if required without the model undergoing any additional training.

When matching a new image, X_1, to these clusters, X_1 and its mirror, X_2, are each input to the network and the mean vector Y of the two output vectors, Y_1 and Y_2, is computed. We calculate a distance metric d from Y to each subject s as shown in Eq. (3), whereby we compute the Euclidean distance between Y and μ_s, the mean of cluster s, proportional to the standard deviation of the cluster, σ_s, in each dimension. This distance is then used to identify the k nearest clusters.

$$d = \left\| \frac{\mu_{s-Y}}{\sigma_s} \right\| \tag{3}$$

We visualise our dataset within the created n-dimensional feature space using the t-distributed Stochastic Neighbour Embedding (t-SNE) method of van der Maaten and Hinton [27], as shown in Fig. 5. This takes the high-dimensional vectors output by our model and creates a low-dimensional embedding of our dataset that allows us to assess how well visually similar samples are being clustered.

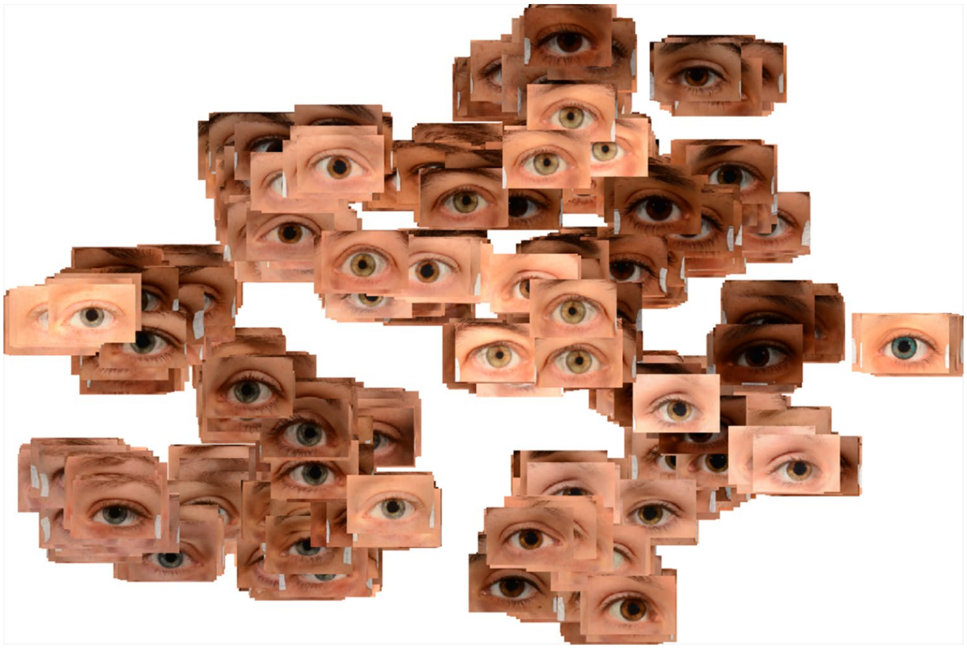

Fig. 5 The embedding of our training dataset by Siamese network output vector visualised using t-SNE [27]

4.3 Generative adversarial network

As well as recommending products, we want to visualise how a chosen product will look when applied to a customer. To facilitate this, we propose a GAN-based approach whereby different networks are trained to simulate the appearance of different products, in our case different mascara formulations.

The architecture of our GAN is shown in Fig. 6. The generator component is based on the u-net architecture of Ronneberger et al. [30]. Originally designed for segmentation, u-net utilises skip connections between corresponding encoder and decoder layers in order to preserve fine details that would otherwise be lost during pooling operations. In our case, this preservation is important as details present in the input images must be preserved if the output is to appear realistic. The final layer in our u-net model is a 1 × 1 learned convolution that converts the final 64-channel feature map into a 3-channel RGB output image.

For the discriminator, we use a VGG16 network [26] with a single output between 0 and 1 that represents the model's confidence that an image is genuine. More recent network designs, such as Inception [31] and ResNet [32], have demonstrated greater accuracy in classification tasks; however, it has been shown that better discriminator performance can adversely impact generator learning [33] and so optimum discriminator accuracy is not a priority. Both models begin training with randomly initialised weights.

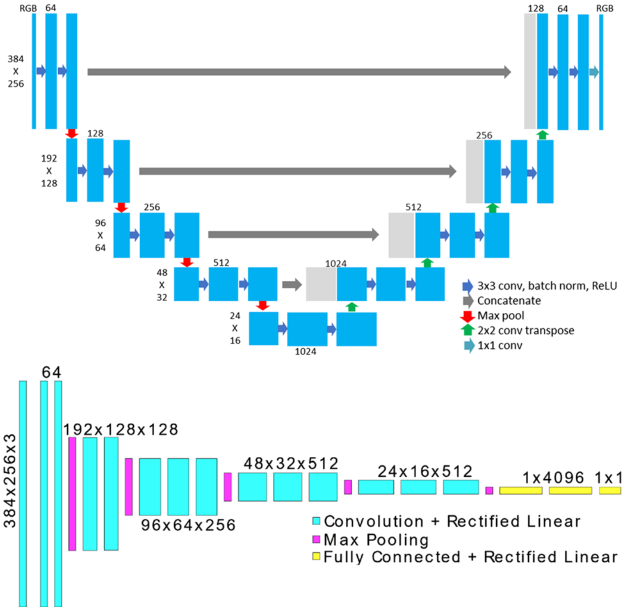

Fig. 6 The configuration of our GAN during training. The generator (top) is based on u-net [30], the discriminator (bottom) is based on VGG16 [26]

The data used to train our GANs consists of two sets, T_0 and T_1, each comprising 720 images. T_0 contains four images of each eye of 90 women with no makeup applied, while T_1 contains images of the same eyes with mascara applied—360 for each of the two formulations used, F_1 (a soap-based formula) and F_2 (a more viscous non-ionic-based formula), with each applied twice to each eye using a different brush to add extra variability to the data.

We train two GANs, GAN_1 and GAN_2, using the F_1 and F_2 imagery, respectively. In each case, we alternate between epochs in which the discriminator and generator are trained. In training the discriminator, we alternate between batches containing real images from T_1 and batches containing images output by the generator. Loss is computed as the mean squared difference between the output and target scalars, which in the case of a real image is 1 and in the case of a generated image is 0, and is backpropagated using stochastic gradient descent with momentum [34].

In generator training, we input batches of images from T_0 and use the sum of two loss functions—pixel loss, l_{pix}, and adversarial loss, l_{adv}—to optimise network weights, again using stochastic gradient descent with momentum. l_{pix} is the mean difference in pixel values between network output and a target image, in this case an image of the same eye with the relevant mascara product applied, and acts to ensure network output takes account of the features of individual input images rather than optimise to generate generic images that fool the discriminator. l_{adv} is the mean squared difference between the output of the discriminator given the output image as input, and a target value of 1, i.e. we want the discriminator to classify the generator output as real.

5 Results

5.1 Siamese clustering

The overall goal of our clustering approach is to create a model capable of matching images of new subjects to existing subjects based on visual similarity; however, this is difficult to assess quantitatively. In order to quantify the performance of our model, we instead input an unseen image from each of the 91 subjects in our training dataset and note whether the correct subject cluster is identified by our matching algorithm within the output k nearest clusters for $k = 1$, $k = 5$ and $k = 10$.

The rationale behind this is that a model capable of accurately matching new subjects will also be capable of matching existing subjects correctly, as the visual similarity should always be high between two images of the same subject. However, as this is not a pure classification task but a metric of similarity, we would not expect the same subject to always return the same single closest cluster, and so by taking account of the nearest 5 and nearest 10 we can better evaluate the model's performance at the task it was designed for.

5.1.1 Effect of vector dimensionality

We evaluate eight trained models with output vectors of dimensionality n of between 8 and 64 in order to observe the effect that output dimensionality has on matching performance. In each case, we tested the model at every 100 training epochs and selected the instance that demonstrated the best performance.

Results are shown in Table 1 and Fig. 7, showing that the best performance was demonstrated by the model with an output vector of $n = 24$. It would appear that an output vector of $n < 24$ is insufficient to encode the information needed to adequately perform this task; however, the additional complexity was introduced when $n > 24$ seems to negatively impact performance. These results are consistent when using $k = 1$, $k = 5$ and $k = 10$ nearest clusters. The top performing configuration, with an output vector of $n = 24$ dimensions, identifies the correct subject within its 10 closest matches for 97% of samples, and as the single closest match for 59% of samples, which considering this is essentially a 91-class problem for which our model has only seen 7 training samples per class, would seem to be quite a good result.

5.1.2 Effect of training duration

For each model, we save the weights after every 100 epochs of training so that the effect of training duration on performance can be assessed. In Table 2 and Fig. 8, results are shown for the training process of our 24-dimension output

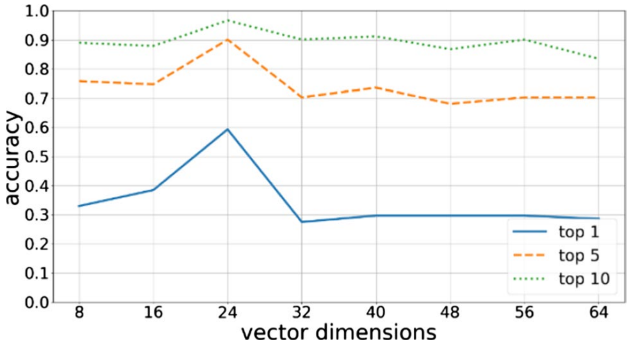

Fig. 7 Plot of results shown in Table 1 demonstrating the effect of output vector dimensionality on matching performance

Table 1 Best performance demonstrated by models of different output vector dimensionality

Dims	8	16	24	32	40	48	56	64
$k = 1$	0.33	0.38	**0.59**	0.27	0.30	0.30	0.30	0.29
$k = 5$	0.76	0.75	**0.90**	0.70	0.74	0.68	0.70	0.70
$k = 10$	0.89	0.88	**0.97**	0.90	0.91	0.87	0.90	0.84

In each case the probability of the correct subject being identified in the top k nearest clusters for $k = 1$, $k = 5$ and $k = 10$ is shown

Bold indicates best performing configuration

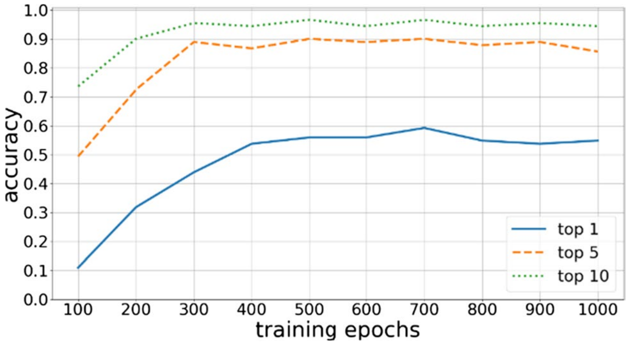

Fig. 8 Plot of the results shown in Table 2 demonstrating how our 24-dimension output model performed after different amounts of training

model, which demonstrated the best overall performance. We can see that the model was able to reliably identify the correct subject within the top 10 nearest clusters very quickly, reaching an accuracy of 0.9 after just 200 epochs. After 300 epochs, top 5 performance appears to be nearly optimised, while top 1 performance continued to slowly improve until 700 training epochs, after which performance degraded slightly.

The best performance is observed at 700 epochs, at which point there are only 3 test samples for which the correct subject is not included in the top 10 nearest clusters.

These 3 failure cases are displayed in Fig. 9 along with a subset of the training images of the correct subject and a subset of the training images of the incorrect closest matched subject. It would appear that a change in lighting between training and test images may have caused the mismatch, as the incorrectly matched images and the input test image both appear slightly darker than the images of the correct subject. In the third example, there seems to be a large amount of variability amongst training samples that may have caused the error; however, unlike the other two examples there is little visual similarity between the input test image and the incorrectly matched subject.

5.1.3 Qualitative evaluation

We have demonstrated the ability of our model to correctly match unseen images to their subjects; however, the aim of this work is to match images of new subjects to the most visually similar of the set of subjects the model has already seen.

Table 2 Performance of our 24-dimension output model during the training process

Epoch	100	200	300	400	500	600	700	800	900	1000
$K=1$	0.11	0.32	0.44	0.54	0.56	0.56	**0.59**	0.55	0.54	0.55
$K=5$	0.49	0.73	0.89	0.87	**0.90**	0.89	**0.90**	0.88	0.89	0.86
$K=10$	0.74	0.90	0.96	0.95	**0.97**	0.95	**0.97**	0.95	0.96	0.95

In each case, the probability of the correct subject being identified in the top k nearest clusters for $k=1$, $k=5$ and $k=10$ is shown

Bold indicates best performing configuration

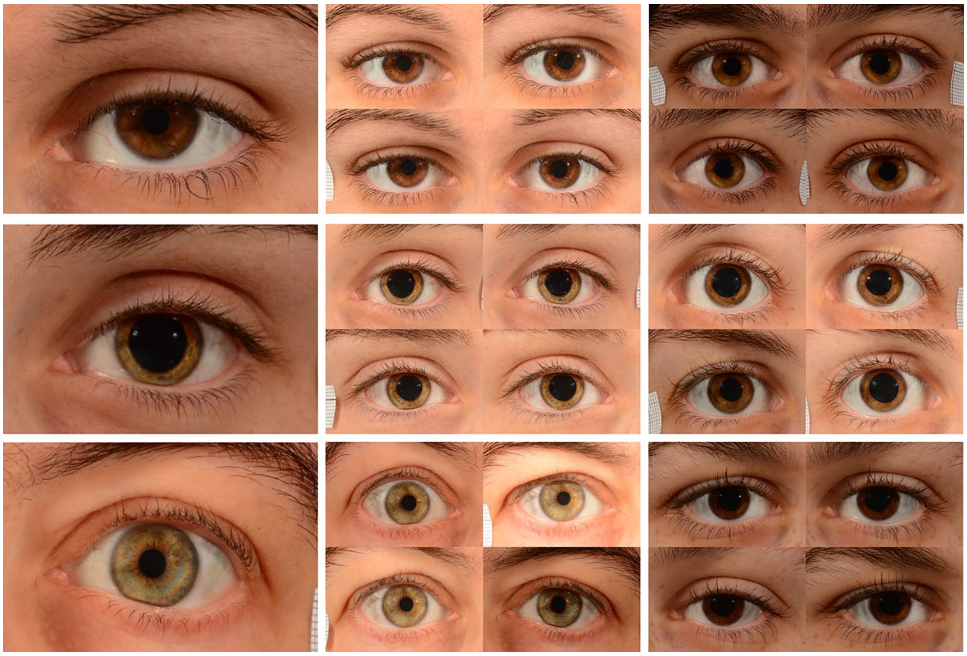

Fig. 9 Three examples where our model failed to identify the correct subject. The left column shows the input test image, in the centre are some of the training images of the correct subject, the right column contains some of the training images of the subject that was incorrectly identified as the closest match

To assess the ability of the model to do this, we input images of subjects not included in the initial training set, captured under the same conditions and cropped using the same HOG/SVM method described in Sect. 4.1. For each input image, we output the top 3 closest matches from the clusters created from our training data. Figure 10 shows some example images along with an image from each of the top 3 nearest clusters. In all cases, we use the model which performed best in our quantitative evaluation, which has an output vector of 24 dimensions and was trained for 700 epochs.

Comparing the input images to those of the closest matched subjects, there appears to be quite a high level of visual similarity in terms of skin tone, eye colour and eye shape, as well as the appearance of eyelashes, eyebrows and eyelid. The two bottom rows of Fig. 10 show the left and right eyes of the same subject, each of which returned the same matches, which is what we would expect if the model is functioning as intended.

5.2 GAN

The training loss curves for both models, Fig. 11, look similar, showing that L_{pix} was very quickly optimised, possibly because the skip connections within the u-net architecture effectively allow the network to 'cheat' at recreation tasks. L_{adv} also optimised quite quickly, however started to slowly increase as the discriminator improved. A high adversarial loss is not necessarily a bad thing however, as a better discriminator generally forces the generator to keep improving the quality of its output.

Testing of the models is performed using a second set of images featuring women not present in the training set, photographed under the same conditions without mascara. Quantifying the performance of a generative model can be difficult, and so in this section we qualitatively assess the output of our model against 3 criteria: similarity of eyelash appearance in generated images to that of genuine T_1 images; similarity of non-eyelash regions of generated images to those of the original T_0 source images; difference between images generated by the two models.

Figure 12 shows outputs from both networks next to their respective T_0 source images and images from T_1 showing the same subject with the real mascara product applied. We can see a clear difference in eyelash length and volume in the generated images that looks plausible based on the genuine mascara images, while changes to non-lash regions of the images are minimal.

Figure 13 compares the output of both models for the same input image. We can see subtle but consistent differences between the two models, such as GAN_1 seeming to add more volume to lashes while GAN_2 appears to improve lash separation, mirroring the qualities of the real products.

6 Conclusions

In this work, we have explored the dual problem of recommending and visualising makeup products based on images of customers. The first part of our approach uses a Siamese network to match a new customer to prior customers whose eyes appear visually similar, so that products favoured by those prior customers may be recommended.

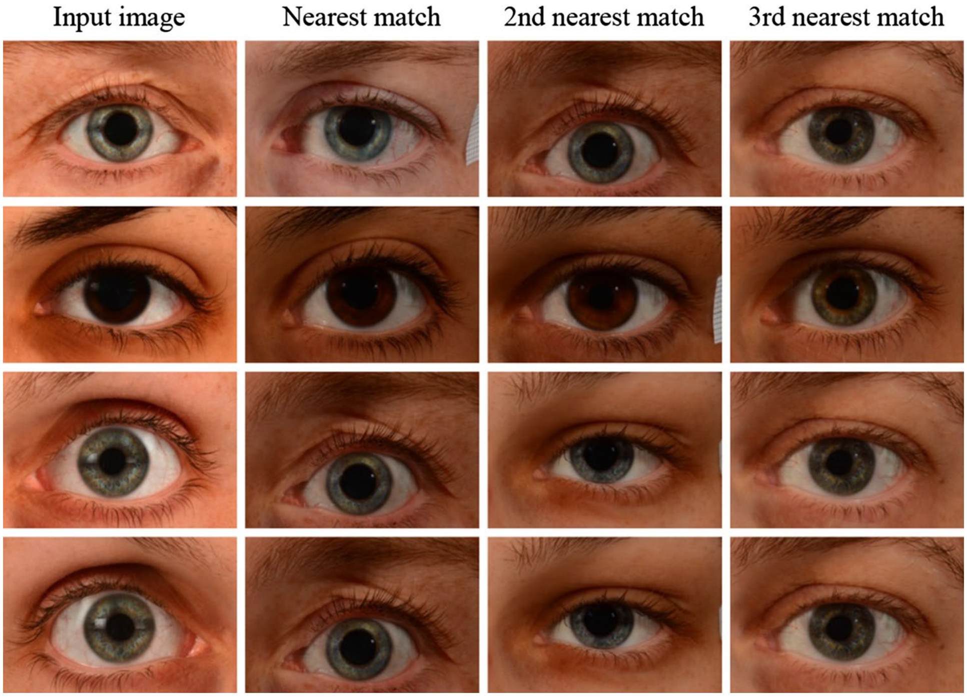

Fig. 10 Some example test images of unseen subjects shown next to images of the 3 closest matches found by our model

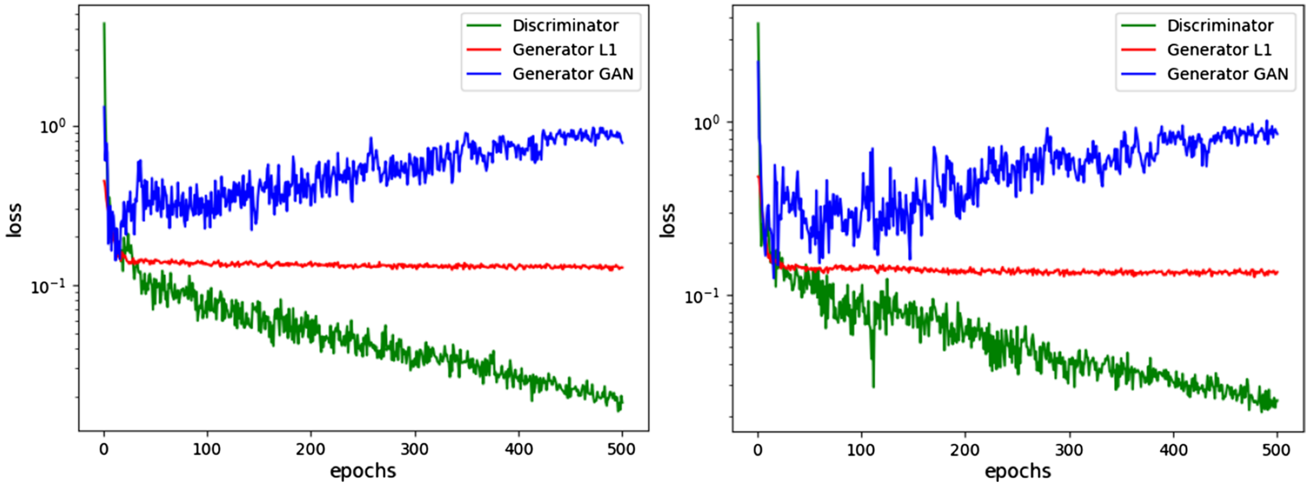

Fig. 11 During training of our two generative networks, GAN_1 (left) and GAN_2 (right), we plot the average discriminator and generator loss at each epoch

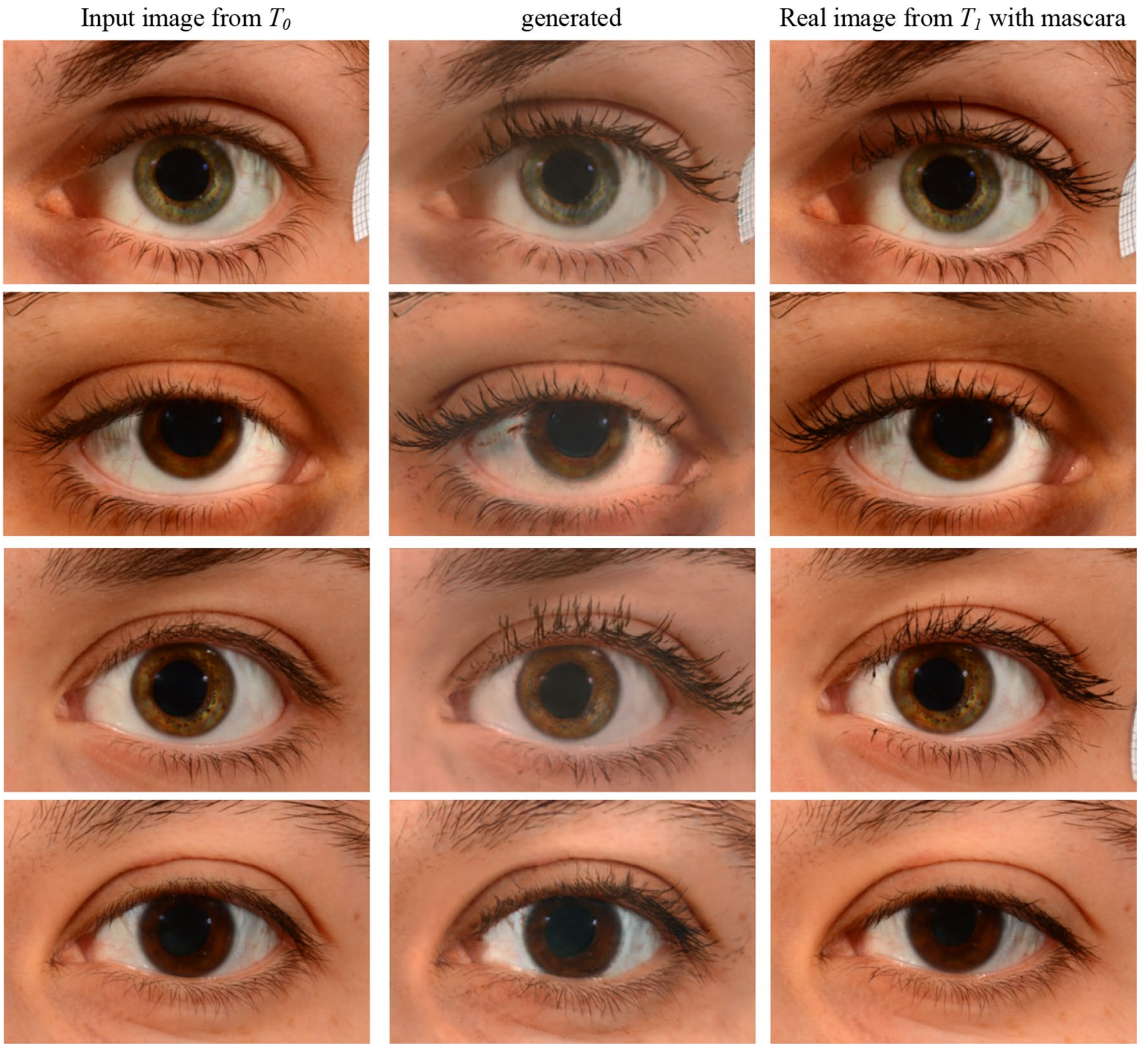

Fig. 12 Example images generated by our two GAN models (centre column). Input image is shown in the left column, while the right column shows images of the same eyes with the real product applied

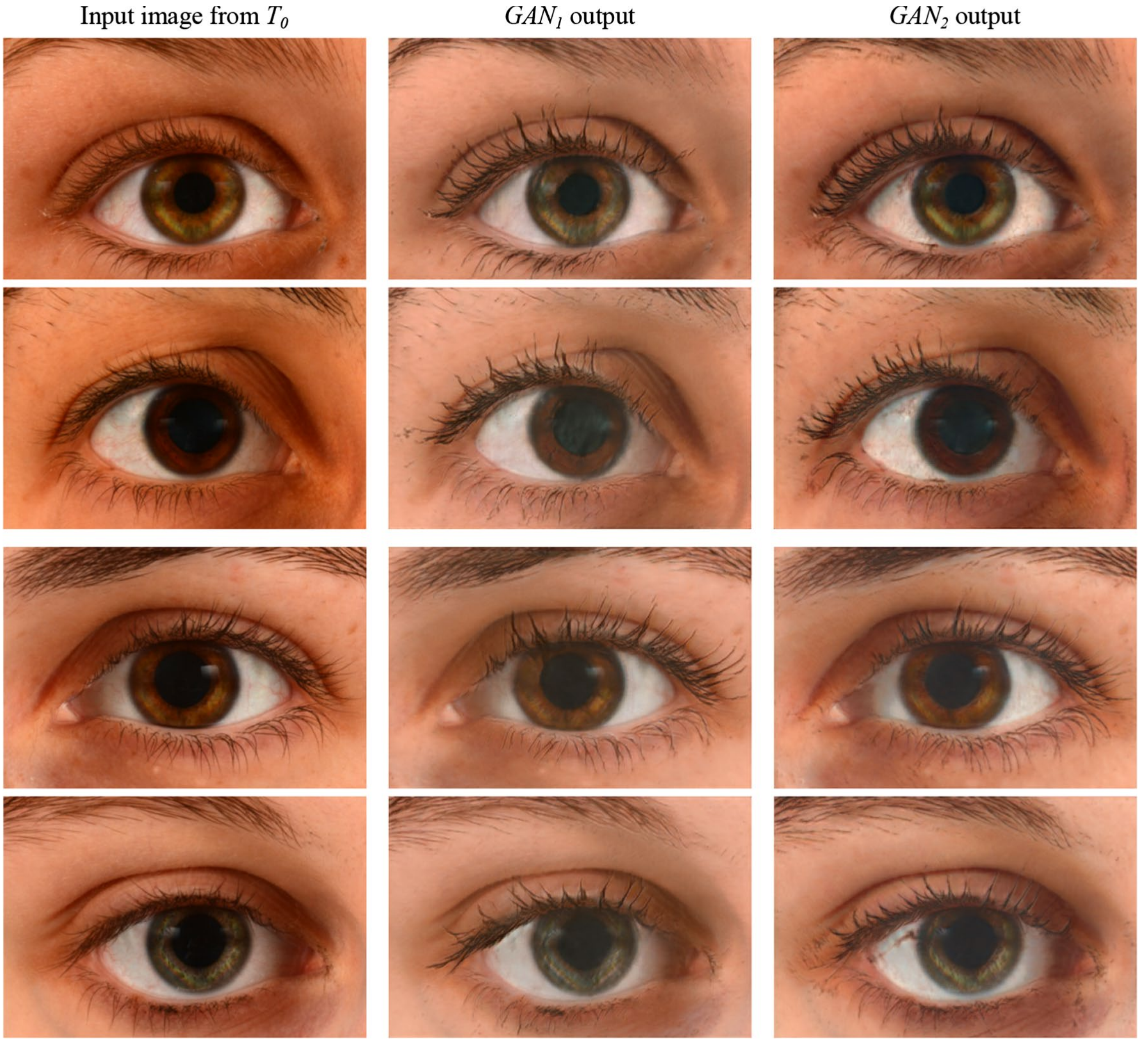

Fig. 13 Example images generated by each of our GAN models from the same input image, demonstrating the subtle variation between products

The second part of our approach uses a generative network trained to visualise how a specific product will look when applied to a customer, helping them to compare and make purchasing decisions without having to apply the product.

We generated our own dataset comprising facial images of 91 women, from which we automatically crop eye regions using an effective HOG descriptor and SVM classifier approach. We then train a Siamese network such that it optimises to cluster together images of the same subject within a high-dimensional embedding, and evaluate its ability to do this using unseen images. We then assess the ability of this trained network to place images of new subjects within the same high-dimensional feature space such that they are near to the stored clusters pertaining to existing subjects that are visually similar.

We evaluate Siamese networks with different output feature vector lengths throughout the training cycle to identify the best-performing configuration, and demonstrate capabilities both in matching new images of seen subjects to the correct clusters, and in matching images of new subjects to clusters that are visually similar.

We then use our dataset, along with images of the same subjects with two different mascara products applied, to train two generative adversarial networks to modify images of naked eyes to realistically portray how they would look with the respective products applied.

We evaluate these networks using images of unseen subjects and qualitatively assess the plausibility of generated images as well as how well the two networks visualise the distinct characteristics of the two products.

References

1. Holder, C.J., Obara, B., Ricketts, S.: Visual siamese clustering for cosmetic product recommendation. In: Asian Conference on Computer Vision (2018)
2. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a "siamese" time delay neural network. In: Advances in Neural Information Processing Systems, pp. 737–744 (1994)
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 (2014)
4. Resnick, P., Varian, H.: Recommender systems. In: Communications of the ACM, vol. 40, Number 3, pp. 56–59 (1997)
5. Gomez-Uribe, C.A., Hunt, N.: The netflix recommender system: algorithms, business value, and innovation. ACM Trans. Manag. Inf. Syst. **6**(4), 13 (2016)
6. Smith, B., Linden, G.: Two decades of recommender systems at Amazon.com. IEEE Internet Comput. **21**(3), 12–18 (2017)
7. Bhatti, N., Baker, H., Chao, H., Clearwater, S., Harville, M., Jain, J., Lyons, N., Marguier, J., Schettino, J., Süsstrunk, S.: Mobile cosmetics advisor: an imaging based mobile service. In: Multimedia on Mobile Devices 2010, vol. 7542, p. 754205. International Society for Optics and Photonics (2010)
8. McAuley, J., Targett, C., Shi, Q., Van Den Hengel, A.: Image-based recommendations on styles and substitutes. In: International ACM SIGIR Conference on Research and Development in Information Retrieval (2015)
9. Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., Wu, Y.: Learning fine-grained image similarity with deep ranking. In: IEEE Conference on Computer Vision and Pattern Recognition (2014)
10. Chopra, S., Hadsell, R., Lecun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: IEEE Conference on Computer Vision and Pattern Recognition (2005)
11. Wang, X., Gupta, A.: Unsupervised learning of visual representations using videos. In: IEEE International Conference on Computer Vision (2015)
12. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: ICML Deep Learning Workshop (2015)
13. Yu, Q., Liu, F., Song, Y.Z., Xiang, T., Hospedales, T.M., Loy, C.C.: Sketch me that shoe. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
14. Akçay, S., Atapour-Abarghouei, A., Breckon, T.P.: Skip-GANomaly: skip connected and adversarially trained encoder-decoder anomaly detection. In: Asian Conference on Computer Vision (2018)
15. Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., Greenspan, H.: GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. Neurocomputing **321**, 321–331 (2018)
16. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A.: Photo-realistic single image super-resolution using a generative adversarial network. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)
17. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: IEEE International Conference on Computer Vision (2017)
18. Li, T., Qian, R., Dong, C., Liu, S., Yan, Q., Zhu, W., Lin, L.: Beautygan: instance-level facial makeup transfer with deep generative adversarial network. In: ACM Multimedia Conference (2018)
19. Liu, S., Ou, X., Qian, R., Wang, W., Cao, X.: Makeup like a superstar: deep localized makeup transfer network. arXiv preprint arXiv:1604.07102 (2016)
20. Oztel, G.Y., Kazan, S.: Virtual makeup application using image processing methods. Int. J. Sci. Eng. Appl. Sci. **50**(5), 401–404 (2015)
21. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 886–893 (2005)
22. Cortes, C., Vapnik, V.: Support vector networks. Mach. Learn. **20**(3), 273–297 (1995)
23. Tintarev, N., Masthoff, J.: Evaluating the effectiveness of explanations for recommender systems. User Model. User-Adapt. Interact. **22**(4–5), 399–439 (2012)
24. Neal, M.: Beyond trust: psychological considerations for recommender systems. In: International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government (2011)
25. Yoo, K.-H., Gretzel, U.: Creating More Credible and Persuasive Recommender Systems: The Influence of Source Characteristics on Recommender System Evaluations. Recommender Systems Handbook, pp. 455–477. Springer, Boston (2011)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
27. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**, 2579–2605 (2008)
28. Chopra, S., Hadsell, R., LeCunn, Y.: Dimensionality reduction by learning an invariant mapping. IEEE Conference on Computer Vision and Pattern Recognition, pp. 539–546 (2005)
29. Kingma, D.P., Ba, J.L.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations (2014)
30. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (2015)
31. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition (2015)
32. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
33. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International Conference on Machine Learning (2017)
34. Qian, N.: On the momentum term in gradient descent learning algorithms. Neural Netw. **12**(1), 145–151 (1999)

Christopher J. Holder received a PhD in Computer Science at Durham University, UK, where he specialised in the application of deep learning techniques in off-road autonomous driving. He has been a researcher at the Institute for Infocomm Research, Singapore, and is currently a postdoctoral researcher at Durham University. His research focuses on the application of deep learning to visual problems.

Stephen Ricketts received an MChem degree in chemistry, followed by a PhD in physical chemistry from Swansea University. He also obtained a diploma from the Society of Cosmetic Scientists. He was employed as a Research Scientist at Haemair Ltd (2008–2009) and Unilever (2009–2017). He is currently employed as a Scientist at Walgreens Boots Alliance. His research interests include cosmetic science, data processing and new method development.

Boguslaw Obara received an MSc in physics from the Jagiellonian University and PhD in Computer Science from the AGH University of Science and Technology, Krakow, Poland. He has been a researcher at the Polish Academy of Sciences (2001–2007), a Fulbright fellow (2006–2007) and a postdoctoral researcher at the University California, USA (2007–2009), and the University of Oxford, UK (2007–2009). He is currently an Associate Professor in Computer Science at Durham University, UK. His research interests include image processing, pattern recognition, computer vision, data science and machine learning techniques applied to a wide range of domains, from biology, medicine and engineering to arts and humanities.