Explorative Visualization and Analysis of a Social Network for Arts: The Case of deviantART

Bart Buter[1], Nick Dijkshoorn[1], Davide Modolo[1], Quang Nguyen[1], Sander van Noort[1], Bart van de Poel[1], Albert Ali Salah[1], Alkim Almila Akdag Salah[2]
[1]Intelligent Systems Laboratory Amsterdam
University of Amsterdam
Amsterdam, The Netherlands
{bart.buter, nick.dijkshoorn, dmodolo, quang.nguyen, alexander.vannoort, bart.vandepoel}@student.uva.nl, a.a.salah@uva.nl
[2] New Media Studies
University of Amsterdam
Amsterdam, The Netherlands
a.a.akdag@uva.nl

***Abstract*— deviantART is one of the leading social online network sites with a focus on user-generated artworks. The website has a rich data archive of around 150 million images uploaded by its 15 million members, making it the largest art platform today. This paper describes an open source toolkit that provides a humanities scholar with necessary computational tools to analyse and visualise deviantART and similar art collections. To this end, we combine tools from different research fields such as network analysis, computer vision, machine learning and data visualisation. The toolkit provides the functionality to extract data about members and their artworks directly from the deviantART website, using network analysis to select key members for further investigation. The chosen members' images are automatically downloaded and annotated with different image features, along with which they can be visualised. The visualisation options offered in the implemented toolkit links images to their originals and can be used to explore and analyse the dataset in an interactive way. The toolkit also features an SVM-based classifier to automatically select features to discriminate artists, artworks and styles, which is hidden from the user behind a simple "suggest features" option.**

Keywords-component; image analysis, feature extraction, classification, deviantART, visual arts, online social networks

I. INTRODUCTION

deviantART[1] (commonly abbreviated as dA) is one of the largest online communities showcasing various forms of user-generated artwork. The website was launched in 2000 and today has over 15 million registered members. The platform is open to everyone and allows non-professionals as well as emerging and established artists to exhibit, promote, market, and share their works within a peer community dedicated to art. All artworks are organised according to a comprehensive category structure that is established by the website and encompasses an impressive range of genres.

dA offers rich information and insight into the dynamics of the online art market [1]. However, unlike its peer Flickr, neither dA's network structure, nor its immense image archive has yet gained the attention of social scientists and humanities scholars. More surprisingly, the visual analytics approach that is so promising for the visualisation of such an environment has not yet been considered. In this paper, we describe an open source toolkit that combines image analysis and data visualisation together to allow exploratory research on the dA community.

dA is a highly interactive and dynamic community where each member has a website to exhibit artwork through the "gallery" feature. Members can explore each other's pages and leave comments on the artwork. Each artist can add other artists' works to his own profile under the feature "favorites", and build a network by adding other members to the watchers list, thus automatically receiving updates (e.g. newly added artwork) about these artists. Network statistics are used to provide information about the popularity of members: basic statistics such as number of pageviews, comments, downloads are listed for each member and piece of art.

The toolkit we describe is designed to help social scientists and humanities scholars analyse dA (and similar online social networks of user-generated artwork). It consists of four components: data collection, feature extraction, classification and visualisation. The source code of a working prototype of the toolkit is provided as a link. For the experiments, a portion of the dA network was scraped, and "interesting" artists were identified by using network analysis tools. These artists' immediate environments were then downloaded, including the gallery, as well as the artworks of his/her watchers'. Our toolkit automatically extracts image features from the downloaded artworks, and through a simple interface lets the user visualise the data. Once the user selects artists and/or categories, the toolkit also applies machine learning techniques to suggest features to the user.

The result renders the most discriminative features of each artist/artwork, besides enabling other artworks to be identified as being a member of, or closely resembling, one of the indicated classes. Through a visualisation application, we ensure that this whole experience is visual and intuitive, presenting the art collections and how they relate to the visual features. The toolkit is thus an attempt to close the "semantic gap" [2], by making an advanced tool available to humanities scholars while keeping the internal mechanics transparent.

This paper is organised as follows. We first summarise our methodology. We briefly describe the network analysis, and

[1] http://www.deviantart.com

detail the image features and classification, followed by the toolkit developed and visualisation choices made. We then report experimental results obtained on a relatively small set of subjects, which nonetheless corresponds to an image database of 1.4GB, and finally conclude.

II. Methodology

The preparation of an analysis tool for dA that is usable to arts and humanities scholars means facing two important challenges.

The first is the selection of a representative subset of artists and artworks. Since the dA collection is vast, exploratory analysis should be guided right from the start by focusing on a smaller portion of the data. A good candidate for this is the "Daily Deviation" selections, which are internally promoted works (on a daily basis). We will present some of our results on a set of artists whose works were featured as daily deviations. For a more global overview of the category structure of the network and the relationship between the artists producing in these categories, as well as for community structure detection, we have pursued a network analysis approach.

The second challenge is the presentation of artists, artworks and categories, so as to allow visually useful and informative comparisons. We have developed a visualisation tool for this purpose, and evaluated a number of image features that could be used to process artworks. For a number of artists or categories, it is possible to visualise the artworks along projected dimensions based on these image features. In the next section, we explain the image analysis part and the subsequent projection.

III. NETWORK ANALYSIS

To obtain to the core of the dA network, we have used a number of assumptions that would weed out most of the members, and help us reach a manageable and relevant set of users. The first heuristic is the subscription status; the paying members of the site are more serious users and have access to more services. These can be automatically determined through scraping. Our first data reduction followed these members, and we thus obtained a network with 100,000 nodes and about 4.5 million links, each representing a user being watched by another user. Watchers get notifications about the activities of the members they are watching. Thus, if a member has a high number of watchers, he/she is able to reach out to a larger audience whenever he/she uploads an artwork.

To get to the core of this sub-network, we have recursively removed nodes (and all connecting edges) that had only a few watchers. Each iteration of this procedure peels off one shell from the peripheries of the network, finally leaving us with a densely-connected graph. We have then analysed this core network within itself and in relation to the larger network. The representatives of the core network also give us a fair picture of the relative structure of categories, and how they are positioned with respect to each other.

IV. Image analysis

Our primary aim in this work is to provide the humanities scholar with interpretable visualisations, by using meaningful image features as guides. Most visualisations are restricted to a small number of dimensions (ideally two, but sometimes more) and to a small number of artworks (of the order of at most hundreds). For this reason, we would like to equip the analysis tool with a broad set of possible image features and let a selection mechanism (which can be the user itself) to select a good subset of these feature for any visualisation task set to the system.

This means that the system needs to be flexible. Artists A and B may be best differentiated by their use of *colour* and *composition*, but maybe artists A and C are best compared by *contrast* and *image entropy*. In this section we discuss the variable features we have implemented for giving the system such flexibility. Since dA contains a plethora of styles (i.e. painting, photography, manga, drawings, etc.), low-level visual features rather than high-level compositional features were preferred to form the basis of the image analysis. One can easily think of extending the proposed system with within-category features that are more targeted. However, we should note that there are over a hundred categories in dA (over two thousand, if subcategories are taken into account).

A. Feature extaction

Visualisation of image sets is not usually performed with the raw image data (the pixel values) due to their high dimensionality. By extracting features from images, we can represent them in a lower dimensional feature-space. Feature extraction thus has several advantages:

- The data become computationally easier to work with due to the lower dimensionality
- By using appropriate features, the data become more suitable for generalisation across images
- Reducing the dimensionality makes it easier to visualise sets of images
- Features can have an intuitive basis, which makes it easier for non-computer-scientists to analyse (sets of) images

In the extraction of image features, a distinction was made between low-level *statistical features* and higher-level *cognitive-based features*.

1) Statistical features

We extract many relatively simple low-level statistical features from the images, each initially input as RGB colour images. The first type is *colour-based* features. Many artists produce collections of art pieces with similar colours and should therefore be (partially) distinguishable using colour-based features. For each of the three RGB channels, as well as each of the three HSV channels, an average and median is calculated over all the values.

The second group of features is composed of the edge to pixel and corner to pixel ratios. Let $\{x_{m,i}\}_{i=1\ldots n}$ be the pixel values of the binary edge-image produced by applying a Canny Edge detector [3] on image *m*. The edge to pixel ratio of image *m* is then computed as:

$$f_{e,m} = \frac{1}{n}\sum_{i=1}^{n} x_{m,i}$$

Similarly, let $\{y_{m,i}\}_{i=1\ldots n}$ be the pixel values in the binary corner image produced by the Harris corner detector. The corner to pixel ratio of image *m* is then computed as:

$$f_{c,m} = \frac{1}{n}\sum_{i=1}^{n} y_{m,i}$$

These two features should be helpful, for instance in distinguishing photographs from other categories such as cartoons and manga. The latter tends to have large plain colour patches, which will decrease the amount of edges and corners. They are also indicative of the type of scenes in photography. A blue sky will not produce many edges or corners, whereas a busy street will.

For the next group of features, image m is converted from RGB to a greyscale intensity image I_m. The average intensity feature $f_{\mu I_m}$, as well as the median intensity $\tilde{I}_m$ is computed. These values provide information about the lightness or darkness of each artwork. To measure the contrast between light and dark areas in images, we use the intensity variance:

$$Var(I_m) = \frac{1}{n}\sum_{i=1}^{n} z_{m,i} .$$

Finally, the entropy of the intensity is calculated as follows:

$$H(I_m) = -\sum_{u=1}^{J} \hat{p}_u \log_2(\hat{p}_u)$$

where $\{\hat{p}_u(z_m)\}_{u=1\ldots J}$ are the histogram bins of the intensity values and are defined as:

$$\hat{p}_u(z_m) = \sum_{i=1}^{n} \delta\left[b(z_{m,i}) - u\right]$$

The function $b: R \rightarrow \{1 \ldots J\}$ returns the index of the bin of the input pixel value in the intensity space and $\delta[g] = 1$ if g=0, and 0 otherwise. This feature characterises the amount of texture in an image. In an earlier version, a Weibull distribution parameterisation for image contrast and texture was also considered [4], but this representation produced some problematic cases for marginal images and was not used afterwards. Fig. 1 shows several intermediate representations of an image for different types of feature. In order to capture localised information, additional compositional features are obtained by dividing each image into a 3x3 grid, and locally deriving the same descriptors.

B. *Cognitively-inspired features*

The human visual system is robust, resilient, and adaptive to different imaging conditions. Models of human visual attention have found application in assessing how humans view images and peruse the image content [5]. The feature integration theory postulates that different feature channels (oriented edges, intensity, colour, etc.) are processed separately in the visual system and later integrated [6]. An influential bottom-up model based on this theory is provided by Itti, Koch and Niebur [7]. Despite its simple architecture, the model is capable of mimicking the properties of early primate vision on complex natural scenes. In this model an input image is decomposed through several pre-attentive feature detection mechanisms. These mechanisms operate in parallel channels over the entire visual scene and conspicuity maps (colour, orientation, and intensity) are created. Each map indicates the presence of a salient cue in its respective channel. After different intermediate steps, the model finally combines the conspicuity maps into a unique saliency map. For further details the reader is referred to [8].

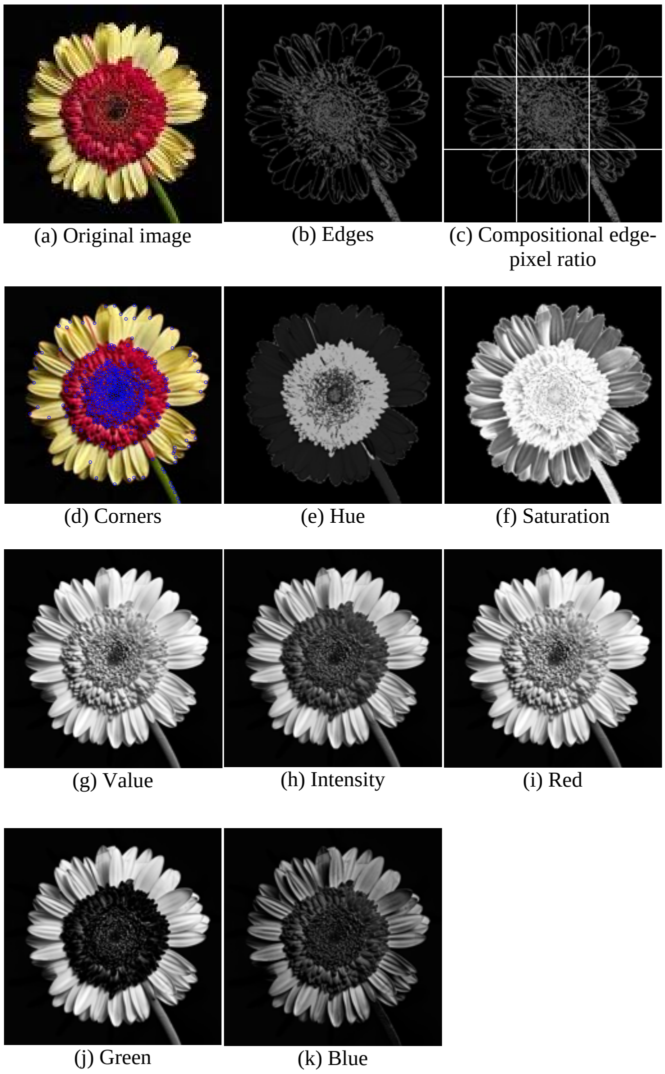

Figure 1. Illustration of different statistical features

We use the saliency map, the individual conspicuity maps, as well as a colour-based skin texture map to derive a set of features that can help our system discriminate between artists and categories. The features that have been extracted from those maps are:

- *Shannon entropy* of the maps
- *Standard deviation* of the distribution of attention in the saliency map
- *Location* of the most salient points (defined as the centres of the most salient regions)
- *Skin intensity* of the skin map

Skin is not a default channel in the saliency model we use, but it has been found to be interesting and useful in dA to distinguish artists and artworks, especially because photography is a major category in dA. Another salient construction is the human face and many visual attention systems explicitly detect faces and make them salient. For this purpose we have used the OpenCV face detection system, which is an Adaboost cascade classifier based on Haar-features [9].

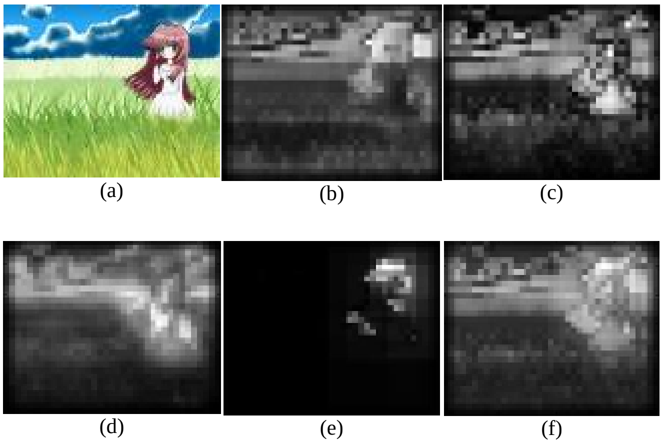

Figure 2. Example feature maps based on the concept of saliency. (f) shows the final saliency map, while (b), (c), (d) and (e) show the conspicuity maps for colour, intensity, orientation and skin, respectively.

The implemented features are shown in Table 1. Part of the statistical feature calculations (i.e. colour-space transformations, edge, corner and face-detection) were performed using openCV [2] [10] to speed up the feature extraction. The saliency features are based on Dirk Walther's Saliency Toolbox[3] [8].

TABLE I. OVERVIEW OF IMPLEMENTED FEATURES. COMPOSITIONAL FEATURES ARE COMPUTED FROM LOCAL REGIONS OF THE IMAGE.

Feature name	Type
RGB	mean, median
HSV	mean, median
Intensity	mean, median, variance, entropy
Edges	edge to pixel ratio
Corners	corner to pixel ratio
Compositional features	all of the above
Face detection	number of faces
Saliency map	standard deviation, entropy, 3 most salient points
Conspicuity maps	entropy, skin ratio

The system designed permits the user to select any two (or more) features, along with which images belonging to a number of artists or categories can be inspected. However, there are a large number of features to select from. For this reason, we endow the system with the capability of selecting features with the maximum separation, given two artists or categories. Moreover, we use machine learning methods to assign scores to each set of features used to describe an artist or a category.

The image features described in the previous section are used in a classification framework. Since most classifiers work better with normalised features, we rescale all data by a min-max normalisation:

$$y_n = \frac{x_n - \max(x_n)}{\max(x_n) - \min(x_n)}$$

where x_n represents the un-normalised values of feature n, and y_n represents the normalised data.

A small set of users were selected from a single day's daily deviation nominees, and all the images in their portfolios were downloaded to constitute the dataset on which the classification experiments were performed. Four different classifiers were contrasted for automatic feature selection:

- *k-Nearest Neighbour* [11] classifies a sample based on the k nearest training examples that are close to it. The Euclidean distance is used to measure the distance between a training sample and the new sample.
- *Naïve Bayes* [12] divides the value range for each feature into n bins. Then it uses feature frequencies to approximate a probability distribution, and uses these to classify an image to the class that gives the maximum posterior probability.
- *Nearest Mean* classifies a sample based on the Euclidean distance to the means of the existing classes.
- *Support Vector Machine* [13] learns a linear decision boundary between two classes in a kernel space, which usually corresponds to a non-linear boundary in the feature space. The distance of stored training samples to the separating hyperplane is maximised during learning.

During the operation of the system, the complete set of features are pre-computed and stored along with thumbnail images for each artwork. This makes it unnecessary to store the dA database locally; the features are of much smaller dimension compared to the original images and the thumbnails can be easily hyperlinked to the original images. Given two classes, represented by two sets of images, or rather two sets of pre-computed features corresponding to these images, the selected classifier will be used to compute a performance score over a set of features. Then two (or more) features will be selected and used in the visualisation.

Instead of the more costly approach of considering all feature combinations, our feature selection scheme proceeds in a greedy manner. It starts by selecting the most informative feature and for each step iteratively adds the next most informative feature to the set of selected features.

The *inter-intra distance* is used as the criterion to decide whether or not a feature is informative. It works by measuring the inner-scatter of a class over a feature and measures it against the scatter of that class around the average of the feature. For a two-class problem the inter-intra distance can be written as:

$$J = \frac{|m_1 - m_2|}{\sqrt{s_1^2 + s_2^2}},$$

where m denotes the class mean, s denotes the standard deviation, and the indices denote the classes. This equation is equivalent to the Fisher criterion [14].

While evaluating the classifiers, every prediction of the classifier is labelled as belonging to one of the four following

[2] http://opencv.willowgarage.com
[3] http://www.saliencytoolbox.net

types, depending on whether or not the classification was correct. This results in a confusion matrix:

	Predicted negative	Predicted positive
Positive	**tp** (true positive)	**fn** (false negative)
Negative	**fp** (false positive)	**tn** (true negative)

From this matrix, we compute the precision P and recall R:

$$P = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fp}}, R = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fn}},$$

as well as the *F-measure*, which is the weighted harmonic mean of precision and recall:

$$F_\beta = \frac{(1 + \beta)^2 PR}{(\beta^2 P) + R}.$$

We use $\beta=1$, which means that precision and recall are evenly weighted.

V. VISUALISATION

In this section we describe the visualization tool that we have implemented to allow the exploration of dA data.

A. Related work

A natural way of visualising a set of images is to extract image features and project them onto a two-dimensional subspace, where each image will be represented as a point. Musha et al. [15] previously developed a visualisation method and an interface for image retrieval, using principal component analysis (PCA). Yang et al. [16] proposed a scalable semantic image browser by applying existing information visualisation techniques to semantic image analysis. Nguyen and Worring have explored several locality-preserving projections in [17]. Our application setting requires us to deal with features with clear meanings. We also aim for a discriminative set of features, rather than treating different classes jointly.

Using multiple visualisations can help us, as the added flexibility increases the users' interaction possibilities with the data. Chen et al. [18] previously used visualisations of images according to features extracted from colour, texture and shape orientation. Schneidewind et al. [19] combined techniques of visual image retrieval and information visualisation to acquire some insight into the extracted feature data, and used multiple representations. Our approach is different from previous work in the literature, as we deal with the requirement of interpretable features. Furthermore we have two levels of annotations per image, the artist that created the image and the image category, respectively. We now describe the software aspects of visualisation.

B. Visualisation software

There are multiple data visualisation applications and toolkits[4] that are able to create visualisations out of the box. However, these only offer generic displays and interactions, which do not capture the dataset in its full potential. Here, we present the visualisation tool we have specifically developed for analysing dA and similar multi-level image collections.

We have implemented several visualisation options to present features on two different levels of abstraction. After offline image analysis, each image is conceived as a point in an n-dimensional feature space.

The first visualisation we use is a *scatter plot* to map the images in a two-dimensional space. Any two features can be used for the axes. Thumbnail versions of the images are used to represent the points, resulting in a low level of abstraction, but with a high level of visualisation. The user can see the actual images on the screen and physically relate to the selected feature dimensions. Fig. 3 shows the scatter plot visualisation technique. The border around each image represents the class (either an artist or a category) to which the image belongs by its colour. If a category is visualised, a representative image set is used, obtained by a k-means clustering, followed by a selection of images closest to each cluster centre. This set is created in an *offline* manner.

The scatter plot is interactive. A single image or all the images belonging to one class can be highlighted, making it easier to recognise patterns. The full version of a miniature image can be displayed to inspect it in more detail. The hyperlinks to original images are maintained.

The scatter plot is limited to displaying only two features at the same time. Our second visualisation option is designed for higher-dimensional feature representations. Fig. 4 shows the *parallel coordinates* visualisation technique [20], a common way of visualising high-dimensional data. The two axes of the scatter plot are now replaced by n vertical parallel lines to represent n features. An image is represented as a polyline with vertices on the parallel axes. The colour of a polyline represents the class to which an image belongs.

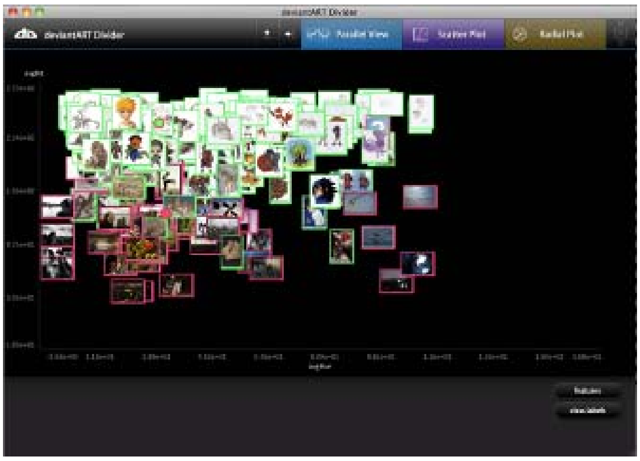

Figure 3. The visualisation application displaying a scatter plot of two artists, using average intensity and average hue as dimensions.

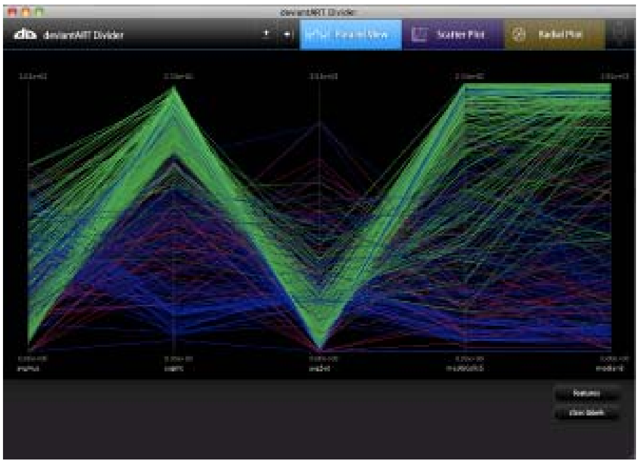

Figure 4. The visualisation application displaying a parallel coordinates plot of three artists and five features.

[4] http://www.wikiviz.org/wiki/Tools

A third visualisation is implemented to show which features are most discriminative for a given artist or category. Fig. 5 shows the *radial plot* visualisation technique used for this purpose. The classes and features are divided into groups, and the line thickness connecting a class with a feature shows the relative discriminativeness of the feature for that class.

C. *The deviantART toolkit*

The complete system, which we call the dA toolkit, consists of four modular components with varying degrees of integration. These modules are for data collection, feature extraction, classification, and visualisation. Here we briefly describe these components from a practical perspective.

Each component writes its output (e.g. image information and features) to an XML file. The first three components are all executed *offline* (i.e. pre-computed), while the visualisation component allows *online* interaction with the images, features and classification results. The offline and online parts of the toolkit are implemented on different platforms.

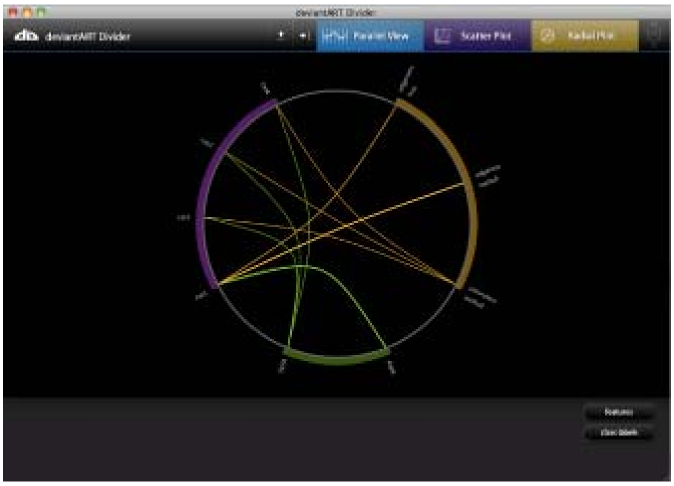

Figure 5. The visualisation application displaying a radial plot that expresses the performance of the classification.

The *data collection* component deals with downloading information and galleries from the dA website and it can easily be replaced to deal with different online collections. The details of data harvesting will be omitted, since they are not essential. For each image, general information such as category, hyperlink, filename, and thumbnail are stored. The full image is downloaded, processed, and discarded. Only the features themselves are stored.

The *feature extraction* is described in Section 3. The classification experiments were implemented in Matlab. For kNN, Naive Bayes, Nearest Mean classifiers and feature selection, the PRTools[5] toolbox was used [21]. LibSVM[6] [22] library was used to implement the SVM classifier because of its high processing speed.

The *visualisation application* and the interface is written in Java programming language. The open source Processing API[7] is used to draw the visualisations. The Processing API contains classes and functions that simplify drawings, animations and interactions in Java [23]. A working prototype of the system is made available as an open source project[8].

VI. EXPERIMENTS

We have collected actual dA data and performed several experiments to optimise parts of the proposed toolbox and to validate its usage. In this section we describe our experimental results.

A. *The deviantART network*

The dA network consists of nearly 15 million (and counting) registered artists. Therefore a sub-network of the dA was extracted made up of professional users. This network formed the basis of our experiments and it contains around 100,000 users. The category structure of dA was derived from this set. As a second step, we found core networks based on high in-degree and out-degree, and reduced the number of users to a thousand. Table 2 describes this network in terms of degree distributions, average path lengths and cluster coefficients. We have downloaded complete galleries for the mixed core, and processed these images with the offline part of the toolbox.

TABLE II. STATISTICS FOR THE PROFESSIONAL AND CORE NETWORKS

Network	Professional artists	Watchers core	Watched core	Mixed core
num. nodes	103663	1701	1471	1099
num. links	4483023	139285	127837	166244
avg. degree	43.25	81.88	86.90	151.27
avg. path length	-	2.15	2.27	2.14
cluster coeff.	-	0.20	0.22	0.20

B. *Image and classification experiments*

For image and classification experiments reported in this paper, we only used the full galleries of 30 artists, which corresponds to about 1.4GB of image data (5324 images). Those artists were selected from the *Daily Deviations*[9] of a random day. Fig. 6 shows the mean images per artist. The dataset is unbalanced (like dA itself), both in terms of images per artist (10 to 500) and images per category. The top five categories in the subset are photography (2244 images), customisation (906), traditional (842), digital art (587) and fan art (239).

Figure 6. Mean image per artist, shown here for 30 artists.

[5] http://prtools.org
[6] http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[7] http://www.processing.org
[8] http://code.google.com/p/ppis-deviantart/
[9] Daily deviations are featured images selected by dA's staff.

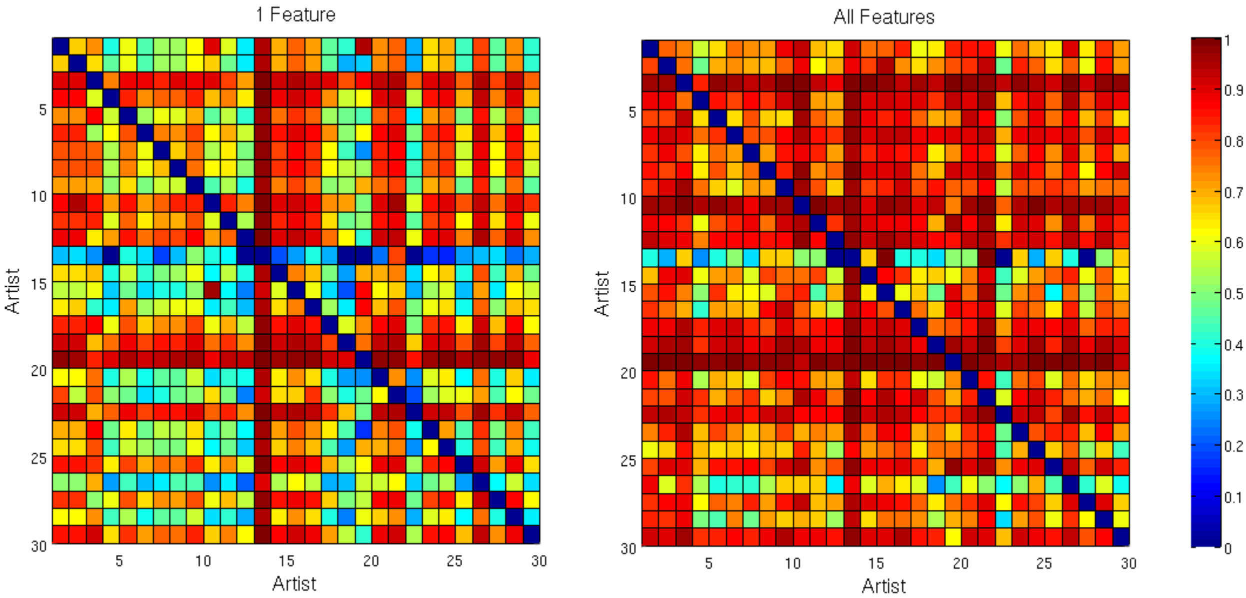

Figure 7. The intensity mapping of the results using the linear SVM in combination with different sets of features. Mapping is divided into 30x30 squares where each square represents the F1-measure of artist x with artist y. The artists are represented on both axes by their numbers. The red colour represents the highest value (F1-measure=1). The left image shows the performance only using the most informative feature according to the selection algorithm, whereas the right image shows the performance of using all features in the mapping.

To evaluate the usefulness of the selected features, we looked at separability under feature projections for combinations of artists. For every artist, the feature selection algorithm uses the inter-intra criterion to select the best features. In a preliminary experiment we tested the kNN, Naive Bayes, Nearest Mean and Linear SVM classifiers, using 5-fold cross validation on 70% of the dataset. The results are shown in Table 3. Linear SVM has the highest mean F_1-measure, and has been selected for the final toolkit. Results of the linear SVM on the test set (the remaining 30%, unseen in the classifier selection phase) are shown in Fig. 7, where colour warmth is mapped to accuracy (red for 1 and blue for 0). Two different feature sets are used with the classifier, the best single feature and all features. We can see that the size of the feature sets used for separate artists is correlated with the performance score. We can also see that most pairs of artists can be separated using only few features. There are even artists who can be separated from all the other artists in the dataset, shown by the columns and rows that are predominantly red. This occurs because those artists use an artistic expression style that is uniquely identifyable when projected onto features that are implemented in the toolkit.

TABLE III. F1-MEASURE FOR EACH CLASSIFIER ON THE TRAINING SET, USING 5-FOLD CROSS-VALIDATION

Classifier	Mean F1-Measure	Stddev F1-Measure
kNN	0.7074	0.1731
Naïve Bayes	0.7897	0.1030
Nearest mean	0.7383	0.1086
Linear SVM	0.8278	0.1450

Artists are more or less consistent in placing the most salient objects in the centre of their works, and features computed from the image centre are more useful than peripheral features. This is also indirectly apparent from Fig. 6.

The most discriminating features obtained by a Borda count [10] over the set of images from one artist-vs-all experiments can be listed as following:

- Entropy of the intensity
- Variance of the intensity
- Centre edge to pixel ratio
- Centre average hue
- Edge to pixel ratio
- Centre average saturation
- Upper-left corner pixel ratio
- Lower-middle edge to pixel ratio
- Entropy of the skin map
- Entropy of the saliency

These results show that the intensity and edge related features are valuable in describing artists. Moreover, there are styles that result in images having many edges (e.g. nature photographs and drawings), but there are also more abstract styles that have very few edges. Other discriminant features are *hue* and *corner pixel ratio*. It also seems that cognitively-inspired features perform best on certain styles. For example, artists that work with people and feature images with a lot of skin, are classified best with the skin map feature.

VII. CONCLUSIONS

This paper describes a toolkit for the analysis of art collections in general, and dA in particular, as well as a research approach that combines online social network analysis with image analysis. We have briefly sketched the network analysis approach we take to reduce the data volume to find interesting users and focus on image processing and visualisation.

[10] A feature that is the most discriminative for a given artist receives three points, the second feature two points, and the third feature one point. The points are summed for all artists.

The toolkit provides multiple levels of functionality to collect information about sets of images and to analyse them through feature extraction and visualisation. While machine learning is used to automatically suggest maximally separating feature dimensions to the user, this is done in a transparent way for the benefit of the intended users, i.e. a humanities scholar exploring the arts collection. A working prototype of the toolkit is provided. The results obtained from a randomly selected user subset have shown that simple feature analysis can be useful to improve the visualisation of galleries of artists. Some image features have been shown to be able to largely separate collections of images and styles. We have used a machine learning approach to automatically suggest these features, as a simple, time-saving extension.

With such a tool, it is possible to answer crucial questions about dA such as "who are the most influential members of dA" [24], "how much do they affect their followers in terms of stylistic changes", or about any collection of artworks, i.e. "could we talk about certain features as being indicative of a particular style or genre?", "is it possible to pinpoint stylistic changes over time in an artist's oeuvre?". To this end, we plan to enhance the toolbox in following directions:

- Temporal information is also present in dA. This dimension needs to be incorporated in order to investigate stylistic changes in the collections over time.
- The number of extracted features can be increased, especially with features that are based on human perception, and with stylistic features. The present set of features is composed of well-known features from image processing and computer vision. Ideally, some semantic notion of the user would drive the feature extraction. An example is the media visualisation of webcomics by Douglass et al., who used overall brightness and publication time to span the axes of a 2D visualisation [25]. Other related works are being performed at Software Studies Initiative at California Institute for Telecommunication and Information[11].
- The emotional impact of an artwork plays a significant role in the artistic creative process. Consequently, the extraction of affective features, such as colour weight, colour activity and colour heat [26] could be useful. The inclusion of texture descriptors would also serve this purpose [27].

References

[1] A. A. Akdag Salah, "The online potential of art creation and dissemination: deviantART as the next art venue," Proc. Electronic Visualisation and the Arts, 2010.

[2] A.W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22, no. 12, pp. 1349–1380, 2002.

[3] J. Canny, "A computational approach to edge detection," Readings in computer vision: issues, problems, principles, and paradigms, p. 184–203, 1987.

[4] J. C. van Gemert, J. Geusebroek, C. J. Veenman, C. G. M. Snoek, and A. W. M. Smeulders, "Robust scene categorization by learning image statistics in context," Proc. CVPR Workshops, p. 105, 2006.

[5] L. Itti and C. Koch, "Computational modelling of visual attention," Nature Reviews Neuroscience, vol. 2, no. 3, pp. 194–203, 2001.

[6] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," Cognitive Psychology, vol. 12, no. 1, pp. 97–136, 1980.

[7] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 20, no. 11, p. 1255, 1998.

[8] D. Walther and C. Koch, "Modeling attention to salient proto objects," Neural Networks, vol. 19, no. 9, pp.1395–1407, 2006.

[9] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proc. CVPR, vol. 1, 2001.

[10] G. Bradski and A. Kaehler, "Learning OpenCV: Computer vision with the OpenCV library," O'Reilly Media, 2008.

[11] F. Korn, N. Sidiropoulos, C. Faloutsos, E. Siegel, and Z. Protopapas, "Fast nearest neighbor search in medical image databases," Proc. Int. Conf. on Very Large Data Bases, pp. 215–226, 1996.

[12] D. Keren, "Recognizing image "style" and activities in video using local features and naive Bayes," Pattern Recognition Letters, vol. 24, no. 16, pp. 2913–2922, 2003.

[13] O. Chapelle, P. Haffner, and V. Vapnik, "SVMs for histogram-based image classification," IEEE Trans. on Neural Networks, vol. 10, no. 5, p.1055, 1999.

[14] W. Malina, "On an extended Fisher criterion for feature selection," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 3, pp. 611–614, 1981.

[15] Y. Musha, Y. Mori, A. Hiroike, and A. Sugimoto, "An interface for visualizing feature space in image retrieval," Machine Vision and Applications, pp. 447–450, 1998.

[16] J. Yang, J. Fan, D. Hubball, Y. Gao, H. Luo, et al., "Semantic image browser: Bridging information visualization with automated intelligent image analysis," IEEE Symp. On Visual Analytics Science And Technology, pp. 191–198, 2006.

[17] G. P. Nguyen and M. Worring, "Optimization of interactive visual similarity-based search," ACM Trans. on Multimedia Computing, Communications, and Applications, vol. 4, no. 1, p. 7, 2008.

[18] C. Chen, G. Gagaudakis, and P. Rosin, "Content-based image visualization," Proc. Int. Conf. on Information Visualization, pp. 13–18, 2000.

[19] A. Schneidewind, P. Neumann, and I. Schmitt, "An approach to visualize image retrieval results," Proc. CVPR Workshops, pp. 148–148, 2004.

[20] G. Andrienko and N. Andrienko, "Constructing parallel coordinates plot for problem solving," Int. Symp. on Smart Graphics, pp. 9–14, 2001.

[21] R. P. W. Duin, "PRTools version 3.0: A Matlab toolbox for pattern recognition," Proc. of SPIE, p. 1331, 2000.

[22] C. C. Chang and C.J. Lin, "LIBSVM: A library for support vector machines," Available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm/, 2001.

[23] B. Fry, "Visualizing Data," CA, USA: O'Reilly Media Inc., 2008.

[24] A. A. Akdag Salah, "Performing curatorial practices in a social network site: The curators of deviantArt," Proc. CHArt Conference, 2010.

[25] J. Douglass, W. Huber, and L. Manovich, "Understanding scanlation: How to read one million fan-translated manga pages," Image & Narrative, vol. 12, no. 1, pp. 190–228, 2011.

[26] L. C. Ou, M. R. Luo, A. Woodcock, and A. Wright, "A study of colour emotion and colour preference. Part I: Colour emotions for single colours," Color Research & Application, vol. 29, no. 3, pp. 232–240, 2004.

[27] M. Lucassen, T. Gevers, and A. Gijsenij, "Adding texture to color: Quantitative analysis of color emotions," European Conf. on Color in Graphics, Imaging and Vision, 2010.

[11] http://lab.softwarestudies.com/