TADA: Toolkit for Analysis of deviantART

Bart Buter, Nick Dijkshoorn, Davide Modolo, Quang Nguyen, Sander van Noort, Bart van de Poel

Abstract—This report describes an explorative research into deviantART. Different research fields have been touched such as network analysis, visual feature extraction, image classification and data visualization. A toolkit has been implemented with the aim of helping researchers to easily explore and analyze the rich data that deviantART offers. It provides functionality to extract data containing images from the devianART website, after which the images are annotated with different types of image features. These image features are used to classify images using different classifiers. Classification experiments have shown that it is possible to discriminate artists, artworks and styles. Furthermore, the toolkit provides a visualization application that can be used to explore and analyze the dataset in an interactive way. Finally, the deviantART network has been analyzed. This is the first attempt to combine online social networks with image analysis.

Index Terms—image analysis, image features, classification, deviantART, art, online social network

1 INTRODUCTION

deviantART[1] (commonly abbreviated as dA) is one of the largest online communities showcasing various forms of user-made artwork. The website was launched in 2000 and has over 13 million registered members. The platform allows emerging and established artists to exhibit, promote, and share their works within a peer community dedicated to art. All artwork is organized according to a comprehensive category structure, including photographs, drawings, manga and short animations. dA is a highly interactive and dynamic community where artists have their own profile containing their artwork. Artists can explore the profiles of other artists and leave comments on their artwork. Each artist can add other artists to the *friends list* to automatically receive updates (e.g. newly added artwork) about these artists. dA provides a very rich dataset, full of interesting information. There has been research into social networks, feature extraction and classification of images and image visualization of large datasets, however there has been no research combining all these aspects. Even more, dA has been researched little in contrast to its peers such as *Flickr*.

This study is meant as *exploratory research* on the dA community, trying to provide answers on art-related questions. As a starting point for our exploratory research we envisioned the following *use case* for a humanities researcher wanting to explore dA. In particular, a database of the dA network is explored with a free available complex network analysis tools such as Pajek[2]. This enables the researcher to identify interesting artwork collections based on their role in the network. These collections can consist of groups of individual artworks, artists' galleries or categories of artworks. All data relevant for the identified collections is retrieved and stored offline. Such data includes the digital artworks themselves, thumbnails representing smaller versions of the artwork, the artist who created the artwork, the date of submission to dA and the category to which the artwork is assigned. The software automatically extracts image features from the artworks which are also stored offline. Through a simple interface a classifier starts to learn the boundary between classes indicated by the researcher. The result of this classifier will enable the researcher to see which features are most discriminative, but will also enable other artworks to be identified as being a member of, or closely resembling one of the indicated classes. A visualization application should make this whole experience visual and intuitive, showing the art collections and how they relate to the features and classifier performance. Moreover, it relates the features, classifier performance and art collections back to the network.

Besides the use case of a humanities researcher doing research, we proposed multiple art-related questions that would be interesting for such a researcher to see answered.

- Can we visualize important aspects of dA?
- Can artists and/or styles be distinguished?
- Are artists influencing each other?
- Do art styles change over time?
- Are there none-artists (dA users that do not produce art, but favorite art from other artists) that are interesting for dA?

To take steps towards making part of the use case reality as well as answering parts of the posed questions, we have embarked on three lines of research. The first was to create a database of a significant part of the network and research the network retrieved. Another line was researching the extraction of visual features and classification of about 5000 images from dA relating to 30 artists. The last pillar of research is the visualization application which enables a visual overview of the fea-

1. http://www.deviantart.com

2. http://vlado.fmf.uni-lj.si/pub/networks/pajek/

tures and artworks for the whole dataset used for feature extraction and classification.

The three lines of research as described above have resulted in a toolkit for dA research. This toolkit has been used to perform experiments which have resulted in identifying cores of interesting artists within the dA network. Other experiments have been conducted using extracted image features with the aim of classifying images into user-based classes.

This report is organized as follows. In section 2.1 we describe the three pillars of research into networks, image features and classifiers (2.2) and the visualization of large datasets (2.3). Section 3 is devoted to the implementation of various parts of the toolkit, as well as their level of integration. In section 4 we describe the experiments which have been conducted and their results. Finally, in section 5 we discuss the future work and in section 6 we draw a conclusion about the research done in this paper.

2 Research areas

2.1 deviantART network analysis

Network analysis has been identified as one of the three main pillars of the research into dA. The purpose of the network analysis is to investigate the properties of the dA network as a whole. Moreover the network is investigated to identify clusters of interest to a humanities researcher. First, previous work will be discussed after which we present a more in depth description of the models used in our investigation.

2.1.1 Previous work

Even though dA presents a rich dataset interesting for the art, social web and computer vision communities, there has been little research into dA. One of the few available academic sources is [1] in which dA is discussed in the context of evaluating a specific peer-review and critiquing application. However, research into art-based social networks exists but concentrates on the photography based network *Flickr*. For instance, favorite markings and their spread through the *Flickr* network have been described in [2]. The structure and growth of the *Flickr* network has been described in [3] and [4]. These articles investigate the user-network with an extra dimension of temperal information about when users first appear and can therefore describe the growth and evolution of the network. The global network model at a macroscopic level is described in [3], while [4] describes the growth at a user or microscopic level. A network model of particular relevance for this project is the small-world model [5] from Watts and Strogatz. Small-world networks inhabit the space between the regular networks or lattices and random networks. Many practical networks have been shown to have a small-world topology, such as the Internet, the power grid and neural networks. For some more examples see [6]. The next section will describe the statistical measures which will enable the identification of small-world networks, as well as some other technical details about these networks.

2.1.2 Proposed approach

In terms of a social network, a large (close to 1) cluster coefficient means that many of a person's friends know each other. A low characteristic path length means that on average two persons are connected through small friends-of-friends chains. Small-world networks are those networks that have a high cluster coefficient and low characteristic path length. Whereas random networks have a low characteristic path and low cluster coefficient and lattices have high cluster coefficients but long characteristic path length.

Formally, let $G = (V, E)$ be a graph, where V is a set of vertices, and E a set of edges between vertices in V, here e_{ij} denotes the edge connecting i and j. The *out-neighborhood* of v_i, $N_i^{out} := \{v_j | e_{ij} \in E\}$, the *in-neighborhood* $N_i^{in} := \{v_j | e_{ji} \in E\}$. The *neighborhood* of v_i, $N_i := N_i^{out} \cup N_i^{in}$. The *degree* of vertex v_i, $k_i := |N_i|$ is the number of vertices in the neighborhood of v_i. A degree can similarly be defined for in and out neighborhoods. The *directional clustering coefficient* for v_i with $k_i > 1$ can now be defined as:

$$C_i := \frac{|\{e_{jk}\}|}{k_i(k_i - 1)}, \tag{1}$$

where $v_j, v_k \in N_i$ and $e_{jk} \in E$, let $n = |V|$ denote the number vertices the *directional clustering coefficient* for G is defined as:

$$C_G := \frac{\sum_{i \in V} C_i}{n}. \tag{2}$$

The *characteristic path length* L_G of graph G is the average shortest path length between vertices in G. Land L_{Gij} denote the length of the shortest path between vertices i and j.

$$L_G := \frac{\sum_{i \in V} \sum_{j \in V \setminus i} L_{Gij}}{n(n-1)}. \tag{3}$$

Finding all shortest paths can be performed using the Floyd-Warshall algorithm [7] having $O(|V|^3)$ complexity.

For networks where $n \gg k \gg ln(n) \gg 1$, the ring lattice will have $L_{lattice} \approx \frac{n}{2k}$ and $C_{lattice} \approx \frac{3}{4}$. A large random network $L_{random} \approx \frac{\ln(n)}{\ln(k)}$ and $C_{random} \approx \frac{k}{n}$. A network is considered small-world when $L_G \approx L_{random}$ and $C_G \gg C_{random}$

To find the core of most connected vertices the findcore algorithm, shown in Algorithm 1, is proposed. This algorithm recursively removes all vertices of degree x from the network, starting at 1. If there are vertices left in the network this is repeated with $x + 1$. Once an empty network is obtained the previous network is deemed the core, because recursively removing nodes of higher degree will inevitably lead to an empty network. The choice of the $degree(j)$ - standard, in-degree, out-degree, or in+out-degree - facilitates in finding cores with different characteristics.

Algorithm 1 FINDCORE($Network$)

Require: A Network $G = (V, E)$.
Ensure: The core of network G and the degree x at which the core is found.

$x \leftarrow 1$
while $|V| > 0$ **do**
  $Gprevious \leftarrow G$
  while $|\{j \in V | degree(j) < x\}| > 0$ **do**
    $G \leftarrow removeNodes(G, \{j \in V | degree(j) < x\})$
  end while
  $i \leftarrow x + 1$
end while
return $Gprevious, x - 1$

2.2 Feature extraction and classification

Computer vision is an important and maturing engineering science. It underpins an increasing variety of applications that require the acquisition, analysis and interpretation of visual information.

2.2.1 Feature extraction

When working with images, it is usually not possible to work with the raw image data (the pixel values) due to its high dimensionality. By extracting features from images, they can be represented in a lower dimensional feature-space. This feature extraction process has several advantages:

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

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

2.2.1.1 *Statistical features*: Many relatively simple low-level statistical features were extracted from the images. The first type of statistical features used are color-based features, which capture the color-usage in the artwork. Many artists produce collections of art pieces with similar colors, and should therefore be (partially) distinguishable using color-based features. For each of the three RGB channels, an average and median is calculated over all the channel values. Let $\{\mathbf{x}_{m,i,c}\}_{i=1...n}$ be the pixel values for image m in color channel $c \in \{R, G, B\}$. The average in channel c of image m is given by:

$$\mu_c(\mathbf{x}_m) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_{m,i,c} \tag{4}$$

The median in channel c is given by

$$\tilde{\mathbf{x}}_{m,c} = \mathbf{x'}_{m,k,c} \tag{5}$$

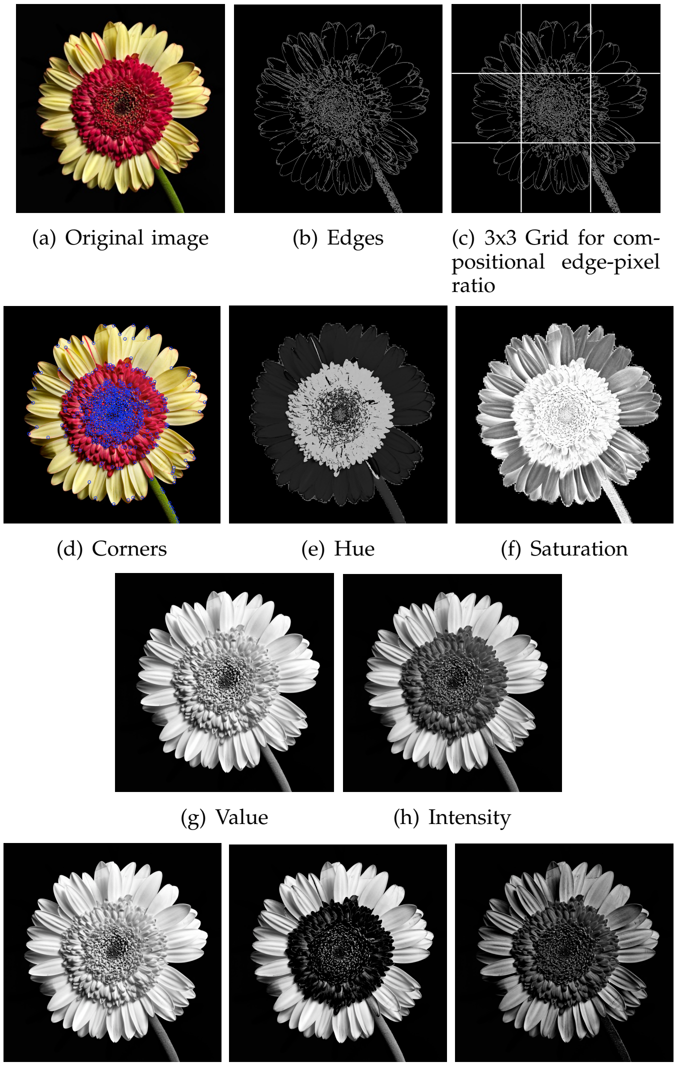
where $\{\mathbf{x'}_{m,i,c}\}_{i=1...n}$ are the sorted pixel values of channel c and $k = \text{round}(n/2)$. The image is also converted into the HSV color space, from which the average and median is extracted for each channel as defined in equations 4 and 5. The Hue channel is given by:

$$H_{m,i} = \left\{ \begin{array}{ll} 0 & \text{if } C_{m,i} = 0; \\ 60\left(\frac{G_{m,i} - B_{m,i}}{C_{m,i}} \text{mod} 6\right) & \text{if } M_{m,i} = R_{m,i}; \\ 60\left(\frac{B_{m,i} - R_{m,i}}{C_{m,i}} + 2\right) & \text{if } M_{m,i} = G_{m,i}; \\ 60\left(\frac{R_{m,i} - G_{m,i}}{C_{m,i}} + 4\right) & \text{if } M_{m,i} = B_{m,i}; \end{array} \right\}$$

Where $M_{m,i} = \max(R_{m,i}, G_{m,i}, B_{m,i})$ and $C_{m,i} = M - \min(R_{m,i}, G_{m,i}, B_{m,i})$. The value channel is given by $V_{m,i} = M_{m,i}$ and the saturation channel is by $S_{m,i} = \frac{C_{m,i}}{V_{m,i}}$

The second group of features is the edge to pixel and corner to pixel ratio. Let $\{\mathbf{x}_{m,i}\}_{i=1...n}$ be the pixel values of the binary edge-image produced by applying a Canny Edge detector [8] on image m. The edge to pixel ratio of image m is then computed as $f_{e,m} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_{m,i}$. Let $\{\mathbf{y}_{m,i}\}_{i=1...n}$ be the pixel values in the binary corner image produced by a corner detector that are either 1 if the pixel is a corner or 0 otherwise. The corner to pixel ratio of image m is then computed as $f_{c,m} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{y}_{m,i}$. These two features should be helpful in distinguishing photographs from other genres such as cartoons and manga. The latter two tend to have large plain color patches, which will decrease the amount of edges and corners. They are also indicative to 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, the artworks are converted from RGB image m to a greyscale intensity image I_m by taking for each pixel i, a weighted sum of the R, G and B channels: $I_{m,i} = 0.2989R_{m,i} + 0.5870G_{m,i} + 0.1140B_{m,i}$. Let $\{\mathbf{z}_{m,i}\}_{i=1...n}$ be the pixel values of the greyscale intensity image of image m. The average intensity feature is calculated as $f_{\mu_{I_m}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_{m,i}$ and the median intensity as $\tilde{I}_m = \mathbf{z'}_{m,k}$, where $\{\mathbf{z'}_{m,i}\}_{i=1...n}$ are the sorted pixel values and $k = \text{round}(n/2)$. These values give information about the lightness or darkness of artworks. The intensity variance feature is computed as $\text{Var}(I_m) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_{m,i}$, which reacts to the contrast between lightness and darkness in images. 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 $\{\hat{p}_u(\mathbf{z}_m)\}_{u=1...j}$ are the histogram bins of the intensity values and are defined as $\hat{p}_u(\mathbf{z}_m) = \sum_{i=1}^{n} \delta[b(\mathbf{z}_{m,i}) - u]$. The function $b : R \rightarrow \{1 \dots j\}$ returns the index of the bin of the input pixel value in the intensity space and $\delta[g] = 1$ if $g = 0$, otherwise 0. This feature somewhat characterizes the texture in an image[3]. Figure 1 shows the intermediate representations of an image for different types of features.



(a) Original image (b) Edges (c) 3x3 Grid for compositional edge-pixel ratio

(d) Corners (e) Hue (f) Saturation

(g) Value (h) Intensity

(i) Red (j) Green (k) Blue

Fig. 1: Illustration of different statistical features

The features described so far only contain global information about images. In order to capture localized information as well, some of the features described above are also extracted from different regions of the image. The regions of the image are obtained by dividing the image along both dimensions into NxM equal-sized regions. Since feature values will most likely vary from region region, these compositional features should provide valuable additional information about an image. Figure 1(c) shows an example of localized edge-pixel ratios using 3x3 equal-sized regions.

2.2.1.2 *Cognitively-inspired features*: A recent trend in computer vision research in the pursuit of human-like capability is the coupling of cognition and vision into cognitive computer vision. Cognitive computer vision has been introduced with the aim of achieving more robust, resilient, and adaptive computer vision systems by endowing them with a cognitive faculty.

Recent studies have focused on computational models of focal visual attention. Attention has been proven to influence the processing of visual information even in the earliest areas of primate visual cortex. Even more, it has been discovered that the interaction of bottom-up sensory information and top-down attentional influences create an integrated *saliency map*, that can be defined as a topographic representation of relative stimulus strength and behavioral relevance across visual space [9]. This map enables the visual system to integrate large amounts of information because it provides an efficient coding scheme for the potentially most relevant information in the sensory input. An important model based on this theory is provided by Itti, Koch and Niebur [10], [11]. Despite its simple architecture the model is capable of mimicking the properties of primate early vision on complex natural scenes. The model works as follow: an input image is decomposed through several pre-attentive feature detection mechanisms which operate in parallel channels over the entire visual scene, and four conspicuity maps (color, orientation, intensity and skin) are created. After different intermediate steps, the model finally combines the four conspicuity maps into an unique saliency map.

Until now the saliency map have been used as information channel in scene understanding and object recognition. In this research, image features are extracted from the map and used in the classification and visualization task. The features that have been extracted from those maps are: *Shannon entropy* of the five maps, *Standard deviation* of the distribution of attention in the saliency map, *Location* of the most salient points (defined as the centers of the most salient regions) and *Skin intensity* of the skin map. Skin is not a default channel in the Itti's model, but it has been found to be really interesting and useful in dA to distinguish artists and artworks, because there is a major presence of photographers that create nude art.

2.2.2 Classification

An important aspect of this project is the description of images, artists or categories using the features that were extracted. In this approach, classifiers are used to give a performance score to a set of features that is used to either describe an artist or a category.

2.2.2.1 *Data normalization*: Classifiers improve their performance when the dataset is normalized. The extracted features have different value ranges, therefore it is important to rescale all the different ranges to a predefined one.

3. The Weibull distribution for image contrast and texture was also investigated, but rejected because of instabilities in the Weibull toolbox.

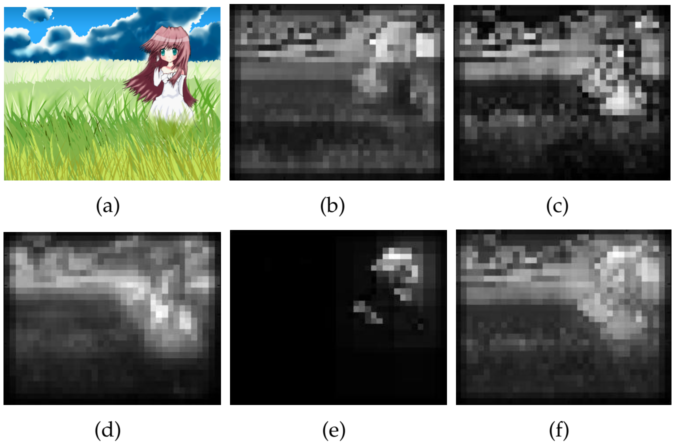


Fig. 2: Example saliency maps. (f) shows the final saliency map, while (b), (c), (d) and (e) show the conspicuity maps for color, intensity, orientation and skin respectively.

$$y_n = \frac{x_n - \max(x_n)}{\max(x_n) - \min(x_n)} * (B - A) \tag{6}$$

Equation 6 shows the *Min/Max normalization* used to normalize the data. In this equation $[A, B]$ represent the predefined range; x_n represents the unnormalized values of feature n and y_n represents the normalized ones.

2.2.2.2 *Classifiers*: Four different classifiers were incorporated to calculate the performance of a set of features. All classifiers were chosen based on their expected performance.

k-Nearest Neighbour [12] classifies an artwork based on the training examples that are close to it. The Euclidian distance is used to measure the distance between a training sample and the artwork.

Naive Bayes [13] divides the value range for each feature in n bins. Then it counts the frequency of a training example for each of the classes in every bin and uses that to classify an artwork belonging to the class that gives the maximum posterior probability.

Nearest Mean classifies an artwork based on the Euclidian distance to the mean of a class. The classifier was chosen because it is a simple model.

Support Vector Machine [14] classifies an artwork using a model learned from training examples. The model represents the training examples by a decision boundary that separates the two classes by maximizing the distance to the decision boundary.

2.2.2.3 *Feature selection*: The classifiers are used to compute the performance score of a set of features. Moreover a feature selection algorithm is used to extract a set of features out of all the features that were precomputed. This is intended to retrieve the smallest set of appropriate features to describe a class.

The feature selection starts by selecting the most informative feature and for each step iteratively adds the next most informative feature to it in a greedy fashion.

$$F := F \cup f, where f : \max J(f_i) \wedge f_i \epsilon F \tag{7}$$

Formula 7 represents this algorithm. In this formula, F represents the entire feature set, f one single feature and J the criterion that is used to define if a feature is informative.

The *inter-intra distance* is used as the criterion to define if 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:

$$J = \frac{|m_1 - m_2|}{\sqrt{(s_1^2 + s_2^2)}} \tag{8}$$

Here m_1 and m_2 are the average mean of class 1 and class 2, s_1 and s_2 are the standard deviations of those classes. This equation is equivalent to the Fisher criterion [15].

2.2.2.4 *Evaluation measures*: An evaluation measure is needed to measure the performance of the classifiers. Every prediction of the classifier is labeled as one of the following four types, depending on whether or not the classification was correct. This results in the confusion matrix:

	predicted positive	predicted negative
positive	**tp** (true positive)	**fp** (false positive)
negative	**fn** (false negative)	**tn** (true negative)

With these standard classification measures two more evaluation measures can be computed. These are precision and recall. *Precision* is defined as the number of relevant artwork correctly classified as positive divided by the total number of positive classified artwork.

$$Precision = \frac{tp}{tp + fp} \tag{9}$$

Recall is defined as the number of relevant artwork correctly classified as positive divided by the total number of positive examples in the dataset.

$$Recall = \frac{tp}{tp + fn} \tag{10}$$

Precision and recall are then used to compute the $F-measure$. This measure is the weighted harmonic mean of the precision and recall and can be defined as:

$$F_\beta = \frac{(1+\beta)^2 (Precision * Recall)}{(\beta^2 * Precision) + Recall} \tag{11}$$

β=1 has been used, which means that precision and recall are evenly weighted.

2.3 Visualization

It is an unfeasible task to manually explore an art dataset by browsing through folders and manually compare image features. Information visualization can be used to visually represent a large-scale dataset, allowing us to see, explore, and understand large amounts of information at once.

2.3.1 Previous work

Image features and information visualization are both research fields that have received great interest. Research that combines both fields is largely coming from the image retrieval field, focusing on efficient methods to retrieve images from (large) image collections.

Musha et al. [16] developed a visualization method and an interface for image retrieval. In their method, principal component analysis is dynamically applied to the image features of the retrieved images in order to determine their eigenspace and the retrieved images are displayed in that space. Statistical experiments showed that their method effectively decentralizes the retrieved images over the two-dimensional space.

Chen et al. [17] compared and analyzed a number of Pathfinder networks of images generated based on low-level image features (color, texture and shape). Salient structures of images are visualized according to features extracted from color, texture and shape orientation.

Schneidewind et al. [18] presented a visualization technique that aims to provide a tool to analyze a mismatch between the users perception and the systems calculation of similarity. They combine techniques of visual image retrieval and information visualization to acquire insight into the extracted feature data. They implemented three visualization techniques to present feature data on three different levels of abstraction: data table, a parallel coordinate plot and a color space plot.

Yang el al. [19] propose a scalable semantic image browser by applying existing information visualization techniques to semantic image analysis. The system consists of multiple visualization components that allows effective high dimensional visualization without dimension reduction. The Multi-Dimensional Scaling (MDS) image view maps image miniatures onto the screen based on their content similarities using a fast MDS algorithm. Interactions are provided to reduce clutter. The Value and Relation content view visually represents the contents of the whole image collection. The correlations among different contents within the image collection, as well as detailed annotations of each image, are visually revealed in this view. The concept-sensitive image content analysis technique was used as automatic annotation engine.

2.3.2 Proposed approach

There are multiple data visualization applications and toolkits[4] that are able to create visualizations out of the box. But these applications and toolkits offer only generic displays and interactions, which do not capture the dataset in its full potential. This disadvantage convinced us to create our own application.

Previous work inspired us to implement multiple visualization techniques to present feature data on two different levels of abstraction. Each image feature represents a dimension in an n-dimensional space. A scatter plot is used to visualize the images in a two-dimensional space. Thumbnail versions of the images are used to represent an image from the dataset, resulting in a low level of abstraction. A parallel coordinates plot is used to visualize a $n \geq 2$ dimensional space. Each image is presented by a polyline, resulting in a higher level of abstraction than the scatter plot.

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

3 TOOLKIT

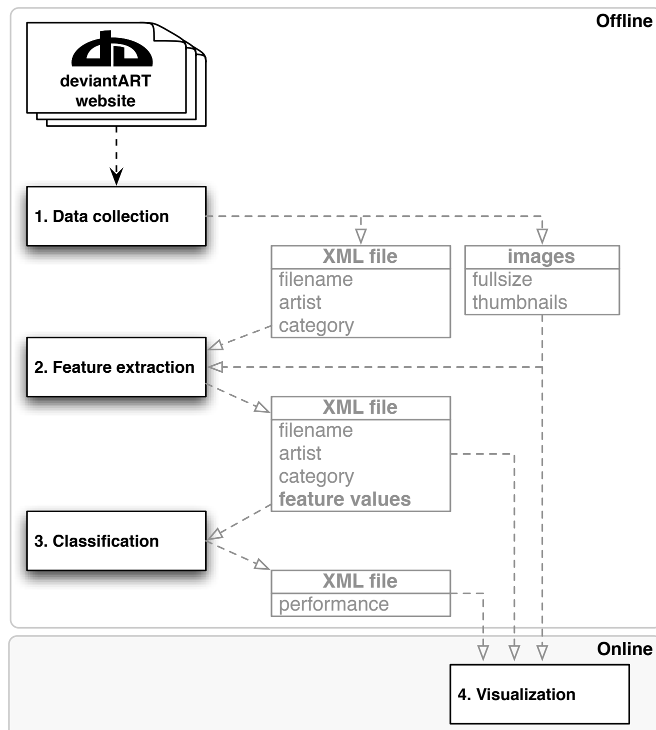


Fig. 3: Interaction between the four components of the toolkit.

The toolkit consists out of four replaceable components with varying degrees of integration. Each component writes its output (e.g. image information and features) to a XML file. The first three components are all executed *offline* (precomputed), while the visualization component allows *online* interaction with the images, features and classification results. Figure 3 provides an overview of all components and the data flow between them.

3.1 Data collection

The data collection component deals with downloading information and galleries from the dA website and it can easily be replaced to deal with different art communities.

dA does not provide a web API to download images and therefore the backend links of the galleries to the RSS XML files were followed to access the image data. For each image general information such as category, deviantART link and filename are stored, and the full image and thumbnails are downloaded.

For the network information collection, the friends pages of the users are parsed. No RSS XML files are provided by deviantART for this information, instead the HTML pages were parsed.

3.2 Feature extraction

The features implemented are show in Table 1. All toolkits and libraries used in this research field, as well as the code to extract the features, are interfaced through the Matlab programming language. Part of the statistical feature calculations were performed using openCV[5] [20] to speed up the feature extraction. openCV was used for color-space transformations, edge, corner and face-detection.

As explained in section 2.2.1.2, the toolkit includes cognitive-inspired features. First a saliency map and 4 conspicuity maps (color, intensity, orientation and skin) are computed and then the features are extracted. To create those maps Itti's model [11] has been used because of its low computational time and the existence of a free toolkit [6].

The XML-processing was done using an open-source XML-toolkit[7].

Feature name	type
RGB	mean, median
RGB compositional	mean, median
HSV	mean, median
HSV compositional	mean, median
Intensity	mean, median, variance, entropy
Intensity compositional	mean, median
Edges	edge to pixel ratio
Edges compositional	edge to pixel ratio
Corners	corner to pixel ratio
Corners compositional	corner to pixel ratio
Face detection	number of faces
Saliency map	std. deviation, entropy, 3 most salient points
Conspicuity maps	entropy, skin ratio

TABLE 1: Overview of implemented features

3.3 Classification

The classification component of the toolkit is implemented in Matlab. For kNN, Naive Bayes, Nearest Mean classifiers and feature selection, PRTools[8] [21] was used. The PRTools toolkit provides functions to create a dataset format that can be used for all the used classifiers.

LibSVM[9] [22] for Matlab was used as SVM classifier because its high processing speed.

5. http://opencv.willowgarage.com
6. http://www.saliencytoolbox.net
7. http://www.mathworks.com/matlabcentral/fileexchange/4278
8. http://prtools.org
9. http://www.csie.ntu.edu.tw/~cjlin/libsvm/

3.4 Visualization

The final component of the toolkit is the visualization application. It is used to present the information that has been gathered by the other components. The collected images are used together with the extracted image features (Figure 3) to visualize the dataset. This provides an effective way to find patterns in the dataset, analyze classification results and filter information. The application combines three different visualization techniques into one application, each of them offering a different look on the dataset.

The application is written in the Java programming language. The open source Processing API[10] is used to draw the visualizations. The Processing API contains classes and functions that simplify drawing, animations and interactions in Java. Processing was an obvious choice, because it has the right combination of cost, ease of use and speed [23].

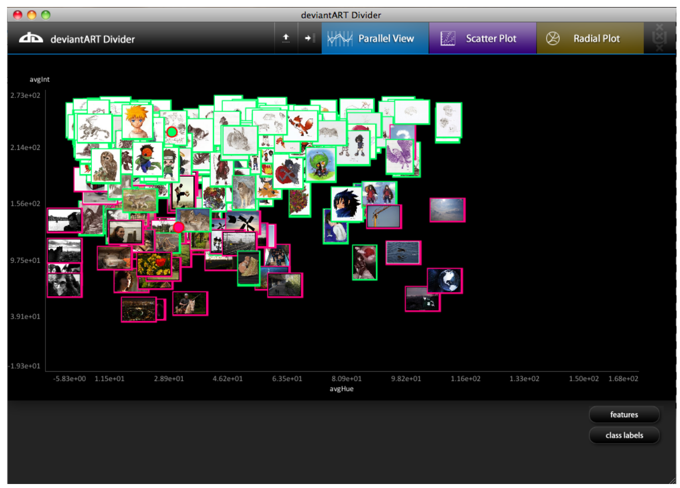


Fig. 4: The visualization application displaying a scatter plot of 2 artists, using average intensity and average hue as dimensions.

Figure 4 shows the *scatter plot* visualization technique. The scatter plot displays values for two variables, which are image features that has been computed by the feature extraction component. The data is displayed as a collection of thumbnail images. Each thumbnail has the value of one feature determining the position on the horizontal axis and the value of the other feature determining the position on the vertical axis. The border around each image represents the class (artists or categories) to which an image belongs. For example, the images with a green border belong to the artist *Kitsunebaka91* and the images with a red pink border belong to the artist *Woekan*. The user has full control over which classes are displayed in the visualization. The users can also control which two features are used as variables on the horizontal and vertical axis of the scatter plot. A single image or all the images belonging to one class can be highlighted, making it easier to recognize patterns. The full version

10. http://www.processing.org

of a miniature image can be displayed to inspect it in more detail.

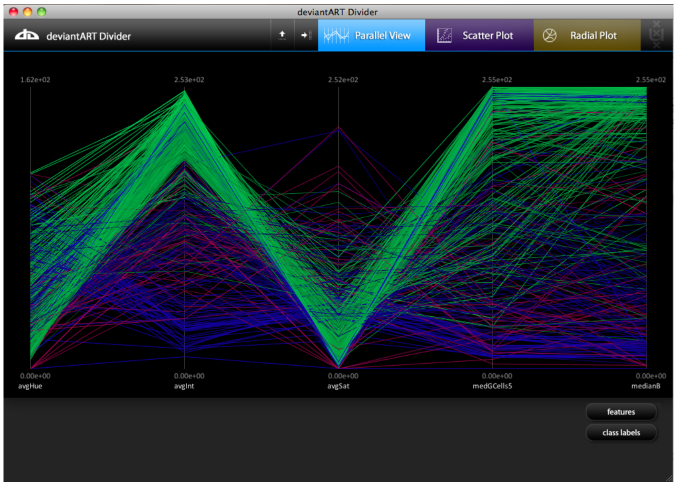


Fig. 5: The visualization application displaying a parallel coordinates plot of 3 artists and 5 features.

The scatter plot is limited to displaying only two features at the same time. Figure 5 shows the *parallel coordinates* visualization technique [24], a common way of visualizing high-dimensional data. This enabled users to visualize beyond two features at the same time. The two axes of the scatter plot are now replaced by n vertical parallel lines to represent n features (n-dimensional space). An image is represented as a polyline with vertices on the parallel axes. The color of a polyline represents the class (artist, category) to which an image belongs.

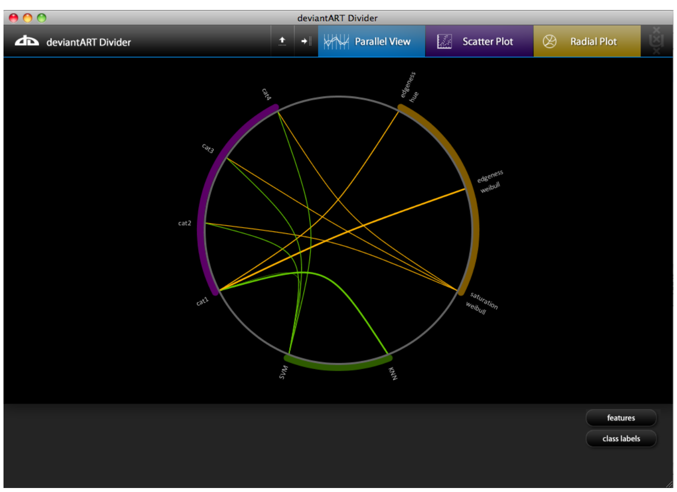


Fig. 6: The visualization application displaying a radial plot that expresses the performance of the classification.

Figure 6 shows the *radial plot* visualization technique. This visualization is not used to visualize the dataset, but to display the performance of the classification, e.g. how a certain feature performs on separating an artist from the other artists. The circle is split into three regions (variables): artists or categories (purple), features (yellow), classifiers (green). The thickness of a line between two nodes expresses the performance of the classification when both nodes are used together, i.e., thicker lines means better performance. For example, a thick line between artist X and feature Y, means that feature Y is a good feature to separate artist X from the other artists.

4 Experiments

4.1 Network experiments

The network experiments have been performed to analyze the dA network and establish global characteristics of the network, but also to find interesting sub-networks of artists and artworks which are of particular interest to investigate.

The dA network consists of 13 million registered artists, and therefore doing a full analysis of the the whole network was not possible within the time-scope of the project. Therefore a network of the dA watchers functionality[11] was extracted for all except casual artists (distinguished with a *tilde* ($\sim$)). This network formed the basis of our network experiments.

From this large network of around 100000 professional artists three core-networks were extracted using the *FindCore* algorithm, described in Algorithm 1. The three cores correspond to three types of the $degree(j)$ function in that algorithm, namely: in-degree, out-degree and in+out-degree. This has resulted in a core of artists that are *power-watchers*, a core with *popular-watched* artists and last a mix of both.

The results of the network experiments are presented in Table 2, Table 3 as well as Figure 7. For all networks, statistics were extracted. Unfortunately for the large professional artists network, statistics such as the characteristic path length were not within the time-scope of this project.

Network	prof. artists	watchers core	watched core	mixed core
no. nodes (n)	103663	1701	1471	1099
no. links	4483023	139285	127837	166244
avg. degree (k)	43.25	81.88	86.90	151.27
findcore max x	-	43	44	185
L_G	-	2.15	2.27	2.14
L_{random}	-	1.69	1.63	1.40
C_G	-	0.20	0.22	0.20
C_{random}	-	0.048	0.059	0.140

TABLE 2: Network statistics

core Network co-occurrences	watchers core	watched core	mixed core
watchers core	1701	541	54
watched core	541	1471	286
mixed core	54	286	1099

TABLE 3: Core network co-occurrences, there are 14 artists which are in all 3 core networks

Through application of the *FindCore* algorithm cores of interesting artists have been identified. These cores are of

11. A watcher is someone who indicates he likes an artist and would like to receive updates when that artist produces new artwork.

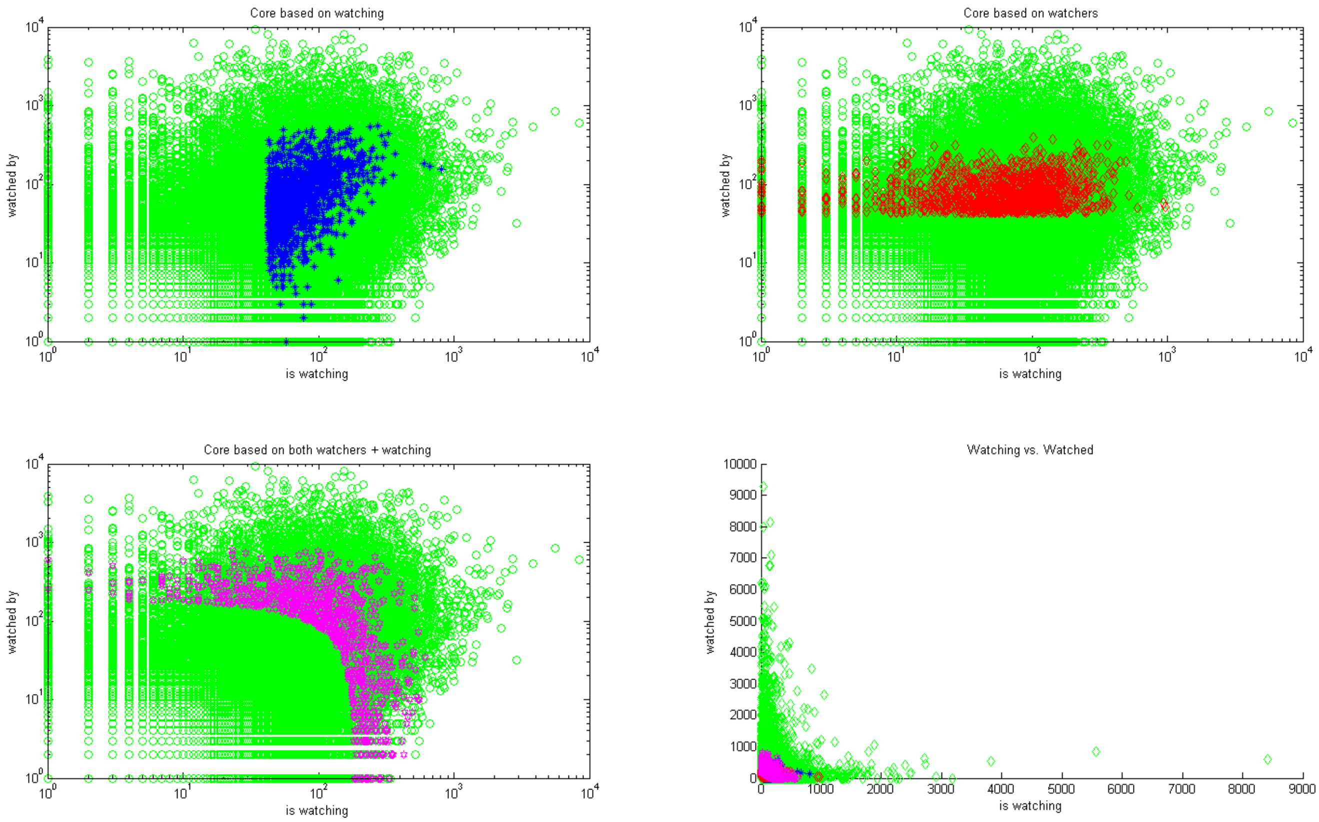


Fig. 7: Core networks, the degree of watching versus being watched. The green dots represent the whole professional network, while the blue represents the watchers-core, red the watched and pink the mixed core, the last sub figure shows the power-law behavior of the degree distributions.

manageable size for full analysis. All core networks have a clustering coefficient larger than a comparable random network, though the mixed core is close to the random network. The characteristic path lengths are longer than their random network equivalents. This is most likely due to the relatively small size of the networks. However, the paths are still significantly smaller than $L_{lattice} = \frac{n}{2k}$. It can been concluded that the cores of dA can be considered small-world networks. Furthermore, the co-occurrences between the watchers and watched network is about $\frac{1}{3}$; this is much lower for the mixed network. It is presumed that this has to do with the FindCore degree $\max x$ which is much larger for the mixed network. Thus the mixed network is distinct from the watchers and watched cores which are more alike.

4.2 Image experiments

For the image experiments, a dataset containing full galleries of 30 artists was collected. Those artists were selected from the *daily deviations*[12] of a random day and includes both premium and non-premium artists. More details about the dataset can be found in Table 4. The dataset is unbalanced (like dA) since some artists only have around 50 images, while other artists have over 500 images. This may influence the performance of the classifications.

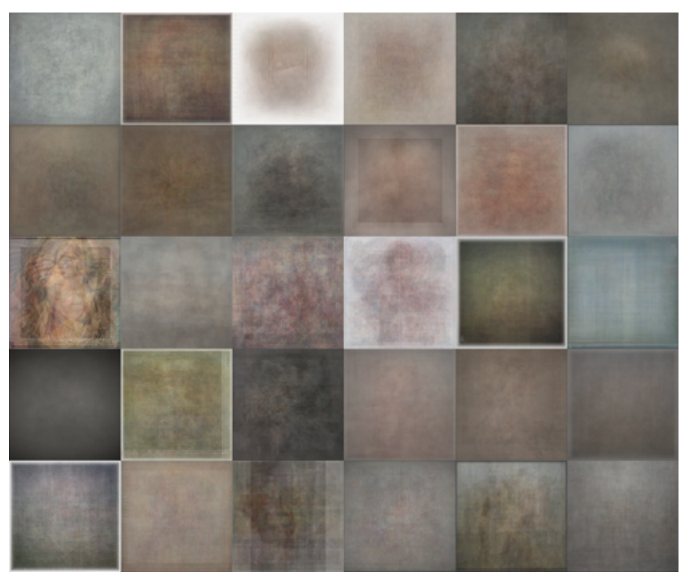


Fig. 8: mean image per artist, left to right, top to bottom, in order of table 4(b): Craniata, K1lgore, Kitsunebaka91, ..., zihnisinir

12. Daily deviations are featured images selected by dA's staff members

(a) General information about the dataset

Total Size	1.4gb
Artists	30
Image count	5324
Average image width	671
Average image height	689
Average image size	250kb
Average image count	171

(b) List of artist with the number of their artworks

Name	count	Name	count
Craniata	61	K1lgore	63
Kitsunebaka91	272	Knuxtiger4	193
LALAax	82	Mallimaakari	151
Mentosik8	110	NEDxfullMOon	116
One-Vox	74	Pierrebfoto	188
Red-Priest-Usada	132	Skarbog	229
Swezzels	9	Udodelig	54
UdonNodu	38	WarrenLouw	51
erroid	146	fediaFedia	303
gsphoto	661	iakobos	47
kamilsmala	39	miss-mosh	307
nyctopterus	68	omega300m	483
sekcyjny	55	stereoflow	166
sujawoto	33	wirestyle	143
woekan	49	zihnisinir	177

(c) Top 5 categories

Category	count
photography	2244
customization	906
traditional	842
digitalart	587
fanart	239

TABLE 4: Dataset statistics

Multiple experiments have been conducted in a preliminary phase, however this paper will only present results for the most significant experiment.

4.2.1 Experimental setup

This experiment focused on determining if an artists is distinguishable from another artist, and which are the best features to separate both. For every artist, the feature selection algorithm uses the inter-intra criterion to select the best features.

Before performing the main experiment, another experiment was conducted to determine the performance of different classifiers. The best performing classifier is than used for the main experiment. The classifiers that have been used are kNN, Naive Bayes, Nearest Mean and Linear SVM. The parameters of the classifiers have been optimized using 5-fold cross validation on 70% of the dataset. The remaining 30% is used as test set for the main experiment. The performance measure used is F_1-measure.

The results are shown in Table 5. Linear SVM has the highest mean F_1-measure score and the second highest standard deviation, and therefore best separates two artists. Based on these scores, linear SVM has been selected for the main experiment.

Classifier	Mean F_1-measure	Stddev F_1-measure
kNN	0.7074	0.1731
Naive Bayes	0.7897	0.1030
Nearest Mean	0.7383	0.1086
Linear SVM	0.8278	0.1450

TABLE 5: The mean F_1-measure and the standard deviation performance score of each optimized classifier on the train set using 5-fold cross-validation

4.2.2 Image experiment results

Results of the trained linear SVM on the test set are shown in Figure 9. The figure shows the *intensity mappings* of four differently sized feature sets (the best 1, 3, 5 and all features). It shows that the size of the feature sets used to separate artists is correlated to the performance score. It also shows that there are artists pairs that can be separated using only few features. There are artists which can be separated from all the other artists in the dataset, shown by the columns and rows that are predominantly red. This happen because those artists use an art style that is unique in the dataset. Furthermore, this result tells that those artists can be described using the features that are implemented in the toolkit.

Table 6 shows the most discriminating features for each artist. Based on these results, some observations can be made. For some artists the mean F_1-measure is very low. This shows that despite the features are the best ones in the entire feature collection, they are not sufficiently describing the artists. The table shows that the *edge to pixel ratio* and *intensity* are valuable features in describing artists. This is very feasible because images can range between being dark and very light. Moreover, there are styles that have many edges (e.g. photographs and drawings), but more abstract ones that have not. Other discriminant features are *hue* and *corner pixel ratio*. Even more, it seems that cognitive-inspired features perform best on certain styles. For example, the artist $Pierrebfoto$ has a lot of nude images and this was exploited by the skin map feature. Another important detail about these results is that the center cell of an image is chosen a lot compared to the other cells. This is actually highly feasible because most of the time in an image, the center contains the most information.

5 Future Work

The toolkit proposed is offering the elementary components to explore part of dA's dataset and to perform some experiments on it. By extending its functionalities many more interesting information could be shown and more experiments could be conducted.

An interesting future investigation could focus on studying the core of the network structure with the aim of answering the following question: *"can the core be representative for the whole network?"* Even more, the network could be made to include more than only *watcher-watching* connections, for example *hierarchical taxonomy*

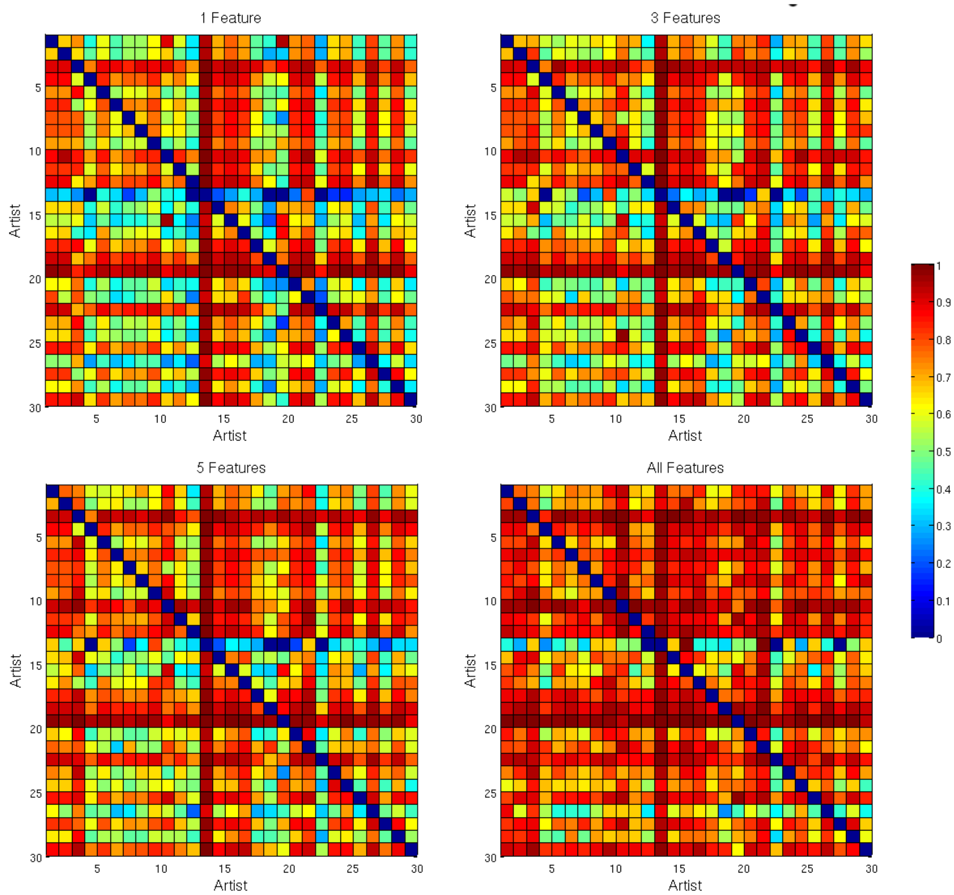


Fig. 9: The intensity mapping of the results using the linear SVM in combination with different sets of features. The mapping is divided into 30x30 squares where each square represents the F_1-measure of artist x with artist y. The artists are represented on both the x and the y axis by numbers. Table 6 shows which artists are corresponding to which number. Also the red color represents the highest value(F_1-measure=1). In the upperleft mapping, the performance using only the most informative feature according to the selection algorithm is mapped. In the upperright mapping, the performance of using the 3 most informative features are mapped. In the lowerleft mapping, the performance of using the 5 most informative features are mapped. In the lowerright mapping, the performance of using all the features are mapped.

connections. Moreover including all artworks and their connections to their users and classes within the taxonomy would enrich the network. An other attracting question concerning the network that could be answered in the future is: *"how the hierarchical artwork classification relates to sub-networks in the core of artists producing artwork in a particular class?"*. dA users describe the existence of hubs: artists who have obtained popularity not by producing original art but as a hub linking interesting artwork through favorites. Future research could try to identify such types of artists based on their behavior in the network. These hubs could be found by relating popularity with the amount of links an artists provides to other persons artworks versus links to his original artwork. Last, incorporating temporal information in the user network would enable us to relate our network to the structure and evolution of the *flickr* network as described in [3] and [4].

It would also be interesting to increase the number of extracted features, especially features that are based on the analysis of human perception. The emotional impact of an artwork plays a significant role in the artistic creative process, and therefore the extraction of emotion-inspired features, such as color weight, color activity and color heat [25] could help the classifier to better distinguish between classes. Moreover, the inclusion of texture would give an even deeper emotional descriptor [26].

Even more, incorporating time into the toolkit would be a great addition, allowing experiments to investigate changes in art over time.

The visualization currently lacks advanced interactions to reduces clutter. It would be nice to incorporate more advanced interactions like zooming and dynamic scaling. Furthermore, the automated communication between the visualization and the other components of the toolkit could be improved. This will allow the user to dynamically expand the dataset by performing an action in the visualization application. Communication between the visualization and the classifier can for instance be used to integrate classification inside the visualization application, allowing its usage without requiring technical knowledge.

No.	Name	μ F_1-measure	σ F_1-measure	Most defining features
1	Craniata	0.6314	0.1864	Center average Hue, Center corner pixel ratio, Center edge to pixel ratio
2	K1lgore	0.6078	0.1813	Entropy of the Intensity, Variance of the Intensity, Edge to pixel ratio
3	Kitsunebaka91	0.8827	0.1756	Center edge to pixel ratio, Entropy of the Intensity, Edge to pixel ratio
4	Knuxtiger4	0.7751	0.1822	Variance of the Intensity, Entropy of the Intensity, Center edge to pixel ratio
5	LALAax	0.6498	0.1902	Edgeratio, Entropy of the Intensity, Variance of the Intensity
6	Mallimaakari	0.7548	0.1824	Variance of the Intensity, Center edge to pixel ratio, Center average Hue
7	Mentosik8	0.6807	0.1876	Entropy of the Intensity, Variance of the Intensity, Lower-left edge to pixel ratio
8	NEDxfullMOon	0.7071	0.1880	Center edge to pixel ratio, Center avg B, Entropy of the Intensity
9	One-Vox	0.6570	0.1833	Entropy of the Intensity, Edge to pixel ratio, Center edge to pixel ratio
10	Pierrebfoto	0.8375	0.1852	Entropy of the skin map, Center average Hue, Edge to pixel ratio
11	Red-Priest-Usada	0.7376	0.1851	Entropy of the intensity map, Center edge to pixel ratio, Entropy of the saliency map
12	Skarbog	0.7875	0.1869	Entropy of the Intensity, Center average saturation, Lower-right average R
13	Swezzels	0.3296	0.2281	Not enough image to report most defining features
14	Udodelig	0.5798	0.1887	Entropy of the Saliency, Entropy of the Intensity, σ saliency distribution
15	UdonNodu	0.5352	0.2158	Entropy of the saliency map, Variance of the Intensity, Upper-middel average Hue
16	WarrenLouw	0.6235	0.1864	Entropy of the Intensity, median B, Upper-right median Intensity
17	erroid	0.7535	0.1833	Variance of the Intensity, Center edge to pixel ratio, Entropy of the Intensity
18	fediaFedia	0.8270	0.1804	Center edge to pixel ratio, Corner pixel ratio, Variance of the Intensity
19	gsphoto	0.9172	0.1779	Entropy of the Intensity, Variance of the Intensity, Center edge to pixel ratio
20	iakobos	0.5764	0.1899	edge to pixel ratio, Upper-left average saturation, upper-left median G
21	kamilsmala	0.6359	0.1978	Variance of the Intensity, Entropy of the Intensity, middle-left edge to pixel ratio
22	miss-mosh	0.8205	0.1811	upper-left corner pixel ratio, Number of faces, Entropy of the Intensity
23	nyctopterus	0.6324	0.1950	Variance of the Intensity, Center median Hue, Center average Hue
24	sekcyjny	0.5875	0.1983	Entropy of the Intensity, Lower-middle edge to pixel ratio, Entropy of the saliency map
25	stereoflow	0.7576	0.1904	Center average Hue, Variance of the Intensity, Corner pixel ratio
26	sujawoto	0.4752	0.1873	Entropy of the intensity map, Entropy of the Intensity, Lower-middle edge to pixel ratio
27	wirestyle	0.7138	0.1931	Entropy of the intensity map, Variance of the Intensity, Center average B
28	woekan	0.5529	0.1881	Entropy of the Intensity, Variance of the Intensity, Upper-left edge to pixel ratio
29	zihnisinir	0.7632	0.1918	Entropy of the Intensity, Center average saturation, Entropy of the intensity map
30	omega300m	0.8318	0.2403	Entropy of the Intensity, Center edge to pixel ratio, Lower-right average Intensity

TABLE 6: This table first describes the mean F_1-measure and its standard deviation for every artist which was computed by using linear SVM combined with a feature set containing the 5 most informative features that separates an artist pair. It also shows the number of the artist that corresponds to the x/y axis of the intensity mappings found in figure 9. The last column shows the most defining features that can describe the artist.

6 Conclusion

This research started with an inspirational *usecase* and interesting research questions in humanity. Some aspects of the *usecase* have been implemented in the toolkit and some experiments have been conducted, both aimed to answer some of the research questions. This has been the first research into deviantART as well as the first attempt to combine online social networks with image analysis.

The toolkit provides functionality to capture a rich dataset containing images from the deviantART website, after which the images are annotated with different types of image features. These image features are used to classify images using different classifiers. The visualization is used to explore and analyze the dataset.

The research has shown that users can be identified based on the location of their artwork in feature space. Furthermore, some image features have been shown to separate artist collections and styles. The best classification result was achieved using a linear SVM classifier.

Moreover, cores of interesting deviantART artists have been identified by analyzing the network. These cores exhibit small-world topology.

References

[1] B.M. Mccreight. A comparison of peer evaluation: The evaluation app versus deviantart. Master's thesis, Purdue University - College of Technology, 2010.

[2] M. Cha, A. Mislove, and K.P. Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *Proceedings of the 18th international conference on World wide web*, pages 721–730. ACM, 2009.

[3] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, page 617. ACM, 2006.

[4] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. Microscopic evolution of social networks. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 462–470. ACM, 2008.

[5] D.J. Watts and S.H. Strogatz. Collective dynamics of small-worldnetworks. *Nature*, 393(6684):440–442, 1998.

[6] R. Albert and A.L. Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47–97, 2002.

[7] Robert W. Floyd. Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345, 1962.

[8] J. Canny. A computational approach to edge detection. *Readings in computer vision: issues, problems, principles, and paradigms*, 184, 1987.

[9] Stefan Treue. Visual attention: the where, what, how and why of saliency. *Current Opinion in Neurobiology*, pages 428–432, August 2003.

[10] Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature reviews. Neuroscience*, 2(3):194–203, March 2001.

[11] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis, 1998.

[12] F. Korn, N. Sidiropoulos, C. Faloutsos, E. Siegel, and Z. Protopapas. Fast nearest neighbor search in medical image databases. In *Proceedings of the International Conference on Very Large Data Bases*, pages 215–226. Citeseer, 1996.

[13] D. Keren. Recognizing image" style" and activities in video using local features and naive Bayes. *Pattern Recognition Letters*, 24(16):2913–2922, 2003.

[14] O. Chapelle, P. Haffner, and V. Vapnik. SVMs for histogram-

based image classification. *IEEE transactions on Neural Networks*, 10(5):1055, 1999.
[15] W. Malina. On an extended Fisher criterion for feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3:611–614, 1981.
[16] Y. Musha, Y. Mori, A. Hiroike, and A. Sugimoto. An interface for visualizing feature space in image retrieval. *InMachine Vision and Applications, pages447-450*, 1998.
[17] C. Chen, G. Gagaudakis, and P. Rosin. Content-based image visualization. *iv*, page 13, 2000.
[18] A. Schneidewind, P. Neumann, and I. Schmitt. An approach to visualize image retrieval results. In *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pages 148–148, 2004.
[19] J. Yang, J. Fan, D. Hubball, Y. Gao, H. Luo, W. Ribarsky, and M. Ward. Semantic image browser: Bridging information visualization with automated intelligent image analysis. In *2006 IEEE Symposium On Visual Analytics Science And Technology*, pages 191–198, 2006.
[20] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, 2008.
[21] R. P. W. Duin. Prtools version 3.0: A matlab toolbox for pattern recognition. In *Proc. of SPIE*, page 1331, 2000.
[22] C.C. Chang and C.J. Lin. LIBSVM: a library for support vector machines, 2001.
[23] Ben Fry. *Visualizing Data*. O'Reilly Media, Inc, 2008.
[24] G. Andrienko and N. Andrienko. Constructing parallel coordinates plot for problem solving. In *1st International Symposium on Smart Graphics*, pages 9–14, 2001.
[25] 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*, 29(3):232–240, 2004.
[26] M. Lucassen, T. Gevers, and A. Gijsenij. Adding texture to color: Quantitative analysis of color emotions. In *European Conference on Color in Graphics, Imaging and Vision*, 2010.