Auto-Illustrating Poems and Songs with Style

Katharina Schwarz[1], Tamara L. Berg[2], Hendrik P. A. Lensch[1]

[1]University of Tübingen, [2]University of North Carolina

Abstract. We develop an optimization based framework to automatically illustrate poems and songs. Our method is able to produce both semantically relevant and visually coherent illustrations, all while matching a particular user selected visual style. We demonstrate our method on a selection of 200 popular poems and songs collected from the internet and operate on around 14M Flickr images. A user study evaluates variations on our optimization procedure. Finally, we present two applications, identifying textual style, and automatic music video generation.

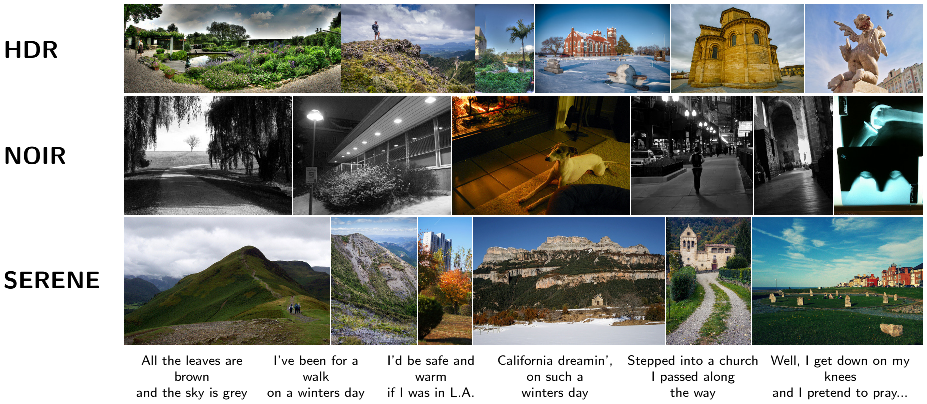

HDR
NOIR
SERENE
All the leaves are brown and the sky is grey
I've been for a walk on a winters day
I'd be safe and warm if I was in L.A.
California dreamin', on such a winters day
Stepped into a church I passed along the way
Well, I get down on my knees and I pretend to pray...

Fig. 1. Automatically generated illustrations of the first text lines (bottom) of the song "The Mamas The Papas - CALIFORNIA DREAMIN" in three different styles (left).

1 Introduction

When an artist creates a poem or song they weave their story carefully, selecting words that produce a vivid visual story in our minds. In this work, we explore the goal of automatically illustrating such creative pieces of art with images. As artists compositions are intended to be highly emotional and lyrical, we aim to select images that are aesthetically pleasing and highly stylistic according to the style of the artwork, e.g., we might illustrate a poem about love with images predicted to be "romantic" but a heavy metal song in "horror" style. Although those kind of texts are quite challenging due to their high level of abstraction, they often display beautiful language, lending themselves well to our goal of auto-illustration with style. Additionally, their nice repetitive structure perfectly suits

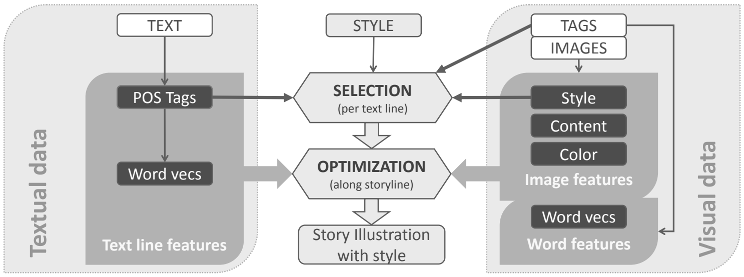


Fig. 2. Overview. Given a large collection of annotated images, we automatically illustrate texts in two steps. First, for each text line, linguistically relevant images are selected that match important words and depict a style. Then, based on this collection, we optimize along the storyline to match style and coherence between selected photos.

our approach. As such texts often deal with a certain theme, we incorporate the global idea as well as allowing for visual adaptions to content changes.

An overview of our processing pipeline is illustrated in Fig 2. Given a large collection of annotated internet photos, we would like to auto-illustrate poems and songs with images reflecting a user-specified style. These illustrations should tell a visual story of the text, matching both the specified image style and demonstrating visual coherence between selected images. This is achieved by first selecting images from the collection that match important words from the input story, depicting the specified style. Suitable candidate images are selected in this manner for each text line. Next, a coherent image sequence is generated to illustrate the story by optimizing style scores and consistency between successive images along the text lines. Consistency is measured using a combination of textual and visual coherence scores related to image content and color.

In order to select good images for illustration, we extract a number of visual and textual features for comparing text to images. In particular, image tags are used in combination with parsed words and word2vec embedding representations [1] to better match not only the syntactic, but also the semantic meaning of images and text. To encourage semantic similarity of image content between subsequent images, the response vector of a deep neural network pre-trained for image classification is utilized, judging content similarity between two images as the distance between the corresponding representations. Finally, the style of each input picture is predicted based on the work of [2]. We use the predictions for 20 different image styles to tell a picture story in a particular illustration style. Each of these criteria can now be used to both assemble a selection of candidate images per text line and then to optimize for consistency along the story using global discrete energy optimization.

The novel combination of considering textual semantic search, content similarity, style classification, and discrete optimization allows us to generate picture story illustrations with controllable style, even for challenging abstract text types such as poems and song lyrics. Some example outputs of our pipeline for different styles are shown in Fig 1. The resulting sequences can easily be synchronized with a song to generate a music video.

2 Related Work

Web-scale Photo Collections: Large community photo collections have been successfully exploited for tasks such as scene reconstruction [3, 4] or scene completion [5]. However, utilizing search engines to query for content usually leads to noisy results due to the weak nature of associated text on the internet. Methods to identify and distill relevant images from these large unstructured data collections are required [6]. Other work has investigated related problems on large web collections, such as extracting storylines depicted in Flickr images of an event like the fourth of July [7], or using YouTube videos to enhance finding an ordering of a photo stream [8]. The latter also uses the images of a photostream to identify keyframes in a user video.

Images to Text: Recently, many researchers have started to explore the relationship between images and the natural language used to describe this imagery. Automatically creating natural language descriptions from images has been presented in several lines of research [9–15]. [9] exploit statistics from parsing large text corpora as well as visual recognition algorithms and output relevant sentences for images. [10] assemble a large filtered set of Flickr images associated with relevant captions to approach the challenge of generating simple image descriptions. Most recent approaches take advantage of deep learning based models and features [11–14], some with models of attention mechanisms [15].

Text to Images: In the opposite direction, namely starting from a textual description to generate or retrieve a visual representation of language has also been investigated in a number of works. [16, 17] provide a graphics engine to render static 3D scenes from natural language descriptions. Whereas [16] allows for user interaction to generate "natural" looking scenes including colors and textures, [17] focuses on automatically resolving spatial relations between 3D objects correctly. An approach for 2D abstract scene generation was presented in [18, 19], modeling scenes using clip-art images produced by workers on Amazon's Mechanical Turk. They are able to learn visual interpretations of simple sentences and automatically illustrate short texts. [20] tackles the Text-to-Image co-reference problem, identifying which visual objects a text refers to, exploiting natural sentential descriptions of RGB-D scenes to improve 3D semantic parsing.

Auto-Illustration: Most relevant to our work, several approaches have been proposed to automatically illustrate text using images. Joshi et al. [21] incorporate an unsupervised ranking scheme to select pictures to illustrate a given story. The Text-to-Video pipeline introduced by [22] assembles a relevant image set using a hierarchical algorithm to retrieve Flickr images with high precision to a given text snippet. Images along the text are chosen by considering RGB color information between candidates of neighboring text parts. The method proposed by [23] learns an association between image sequences and multiple sentences.

Style: Similarly to previous auto-illustration approaches, we also retrieve images that match an input text. However, we also optimize for two additional storyline features: 1) we select images for illustration according to a particular story style, and 2) we attempt to select a visually coherent set of images for illustration. In order to illustrate stories according to a particular style, we rely on previous work for style recognition in images [2]. This approach predicts style using features from a pre-trained multi-layer deep network, fine-tuned to predict image style on 80K Flickr photographs depicting 20 different styles.

3 Feature extraction

In order to support our algorithm we extract a set of features from images and their associated tags as well as from lines of text. The features allow matching both content and style between individual lines of text and images, and between pairs of images selected to illustrate a sequence of text lines.

3.1 Text features

Language features are extracted by applying **parsing** to determine relevant words from the input text while **image tags** are used directly. **Word-vector representations** are generated for both extracted words and image tags.

Parsing. A line l of text T is first analyzed by tokenizing and parsing to determine part of speech (POS) labels for each word [24] and a lemmatizer based on WordNet [25, 24] improves performance. The extracted nouns, verbs, adjectives, and adverbs are gathered in a set $\tau(l)$. POS parsing enables us to select the most relevant words for each matching task. For candidate retrieval, we consider the subset $\tau_{NV}(l) = \{w_1...w_A\} \subseteq \tau(l)$ of extracted nouns and verbs to obtain a broad and large enough image set. Later, the word-vector representations for text lines are computed based on the entire set, $\tau(l)$, to capture additional meaning.

Image Tags. All images considered for illustration have user associated tags. Thus, for each image I we store its associated list of tags $\kappa_I = \{w_1...w_K\}$ in an inverted file table, making it efficient to access all images with tags matching words $\tau_{NV}(l)$ from a text line l.

Word-vector Representation. In addition to directly matching between words, we exploit recent work that maps words to vectors (word2vec) based on a continuous skip-gram model, providing a mapping of phrases into a 300d vector space [26, 1]. The mapping keeps and expresses a large number of precise syntactic and semantic word relationships while compressing semantic similarity. For a single word w, we obtain its word2vec representation $V(w)$ and, for a set of words, we average the vector representations of all words in the set. We calculate word vectors for tags of a text line $\tau(l) = \tau_l$ as V_{τ_l} and for image tags κ_I as V_{κ_I}.

3.2 Image features

Features are extracted from the images to identify **style**, **content**, and for ensuring **color** consistency between selected images.

Style Feature. Estimates for 20 style-classes are extracted using the method of Karayev et al. [2] which classifies image style using a convolutional-neural-network approach. This estimate is used to consider only images that match the specified style, and to ensure consistency between images selected for illustration. We assume that an image I matches a certain style, sty, if its prediction score for this style is greater than 0.5, defining the style constraint as $I_{sty} > 50\%$.

Table 1. Style in YFCC14M. For all 20 provided styles, the table indicates the amount of images within the YFCC14M subset with $I_{sty} > 50\%$ for a certain style. "Detailed" contains the highest amount and half of the styles hold at least more than 300K images.

Style	> 50%	Style	> 50%	Style	> 50%	Style	> 50%	Style	> 50%
Detailed	674K	Noir	352K	Serene	324K	Melancholy	234K	Sunny	201K
Bright	519K	Hazy	335K	Minimal	303K	Long Exposure	227K	Pastel	184K
Horror	498K	Geom. Comp.	334K	HDR	296K	Vintage	224K	Macro	135K
Depth of field	408K	Bokeh	327K	Romantic	259K	Texture	219K	Ethereal	87K

Image Content Feature. To compare the visual content of two images we make use of deep learning results, in particular, the VGG 16-layer model [27]. This deep convolutional network has demonstrated high accuracy on image classification. We use a pre-trained model which has been trained to recognize 1000 classes from the ImageNet Challenge. For our image content representation we use the 4096d features extracted from the first fully-connected network layer.

Color Feature. As color significantly influences our visual impressions of images and is not well represented using pre-trained CNN features, we incorporate a simple RGB color histogram feature for evaluating image similarity. We extract 256 bins per color channel for each image.

4 Corpora

To demonstrate our approach, we make use of publicly available data, crawling a set of 200 famous poems and song lyrics, and use the YFCC100M dataset.

Creative Text Corpus. We assemble a set of creative texts, namely songs and poems to demonstrate our approach. We obtained lyrics to 100 songs from http://www.songlyrics.com. In order to cover different music styles, we crawled the lyrics from the "Top 100 Songs of All Time" category. For poems, the website http://100.best-poems.net provides best poem texts for various categories. To retrieve a broad set across categories, we downloaded the poems in their "Top 100 Poems" category which covers famous poems of all time.

Image Corpus. For generating illustrations, we make use of the "Yahoo Flickr Creative Commons 100 Million Dataset (YFCC100M)" [28] recently published by [29]. It contains about 100 million Flickr images and videos with associated meta-data. We consider the image portion, around 70% of which have textual tags. We select images that are potentially relevant to our text corpus by filtering out images that are not tagged with relevant words, i.e., nouns and verbs appearing in our creative text corpus. Text pre-processing is used to lemmatize, remove stopwords, and select words that occur in both the poems and songs, leaving us with a representative set of about 400 words. The frequency of the resulting words mentioned in the tags reached up to 820079 for the word "music". For around 60 words more than 100000 images do match. Only Flickr images whose tags match at least one of the words in this representative list are selected, leaving us with a subset of 14 million images. Table 1 indicates the amount in our YFCC14M subset with a prediction greater than 50% for each style, typically yielding several hundred thousand images that are likely to depict that style.

5 Selection and Optimization

Given the feature vectors computed for each database image and the similarity measures defined in Sec. 3, the selection process for auto-illustration first computes a suitable candidate set of images matching to each text line. Then, based on those pre-selected candidate image lists, an optimization step estimates the best sequence of images for illustration, maximizing text and style match scores, as well as cohesion in content, color, and style along the illustration.

5.1 Selecting Candidate Images

For every line in a text, a set of candidate images is selected that semantically matches the text line and corresponds to the specified illustration style. Specifically, given the POS analysis, candidate selection is performed for each text line l by including each image $I \in$ YFCC14M into the set of candidate images I_{cand_l}, if the following condition is fulfilled:

$$(\kappa_I \cap \tau_{NV}(l) \neq \emptyset) \wedge (I_{sty} > 50\%) \Rightarrow (I \in I_{cand_l}) \tag{1}$$

This means we select all images whose tags match at least one noun or verb present in the text line and, at the same time, depict the requested illustration style with a style prediction score greater than 50%. Sorting this list according to style score, the top 1000 candidate images per text line form a suitable basis for our optimization phase.

5.2 Storyline Optimization

Given the candidate image sets for each text line, we would like to select a final set of images to illustrate our story that are: 1) semantic relevant to the story, 2) good representatives of the selcted style, and 3) visually coherent along the story line. Fig. 3 visualizes the underlying structure for two successive text parts, each connected with a small subset of potential images for selection. The task of choosing the best image from each subset, while preserving semantic relatedness and visual consistency can be described as a discrete optimization problem with pairwise variables and formulated as an energy with unary and pairwise terms.

Thus, from an assembled and presorted set of images per text line, the goal is to select one image for each position, X_i, of the corresponding text line $i \in \nu$. We formulate this as minimization of an energy function E with image labels for a text line i along all lines ν (nodes) and over consecutive text pairs ε (edges):

$$E(X) = \sum_{i \in \nu} U(X_i) + \sum_{i,j \in \varepsilon} P(X_i, X_j) \tag{2}$$

The unary potential function U measures semantic and stylistic relatedness between a text line and a potential image for illustrating that line. The pairwise consistency terms P describe the interaction potential between pairs of images.

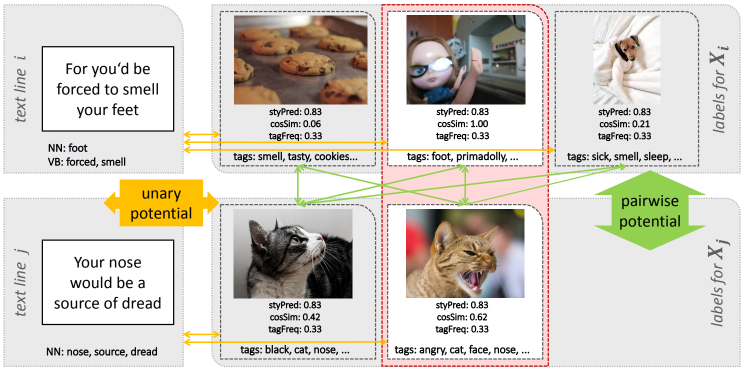


Fig. 3. Optimization structure for two text lines. The images in the middle are finally picked. ("J. Prelutsky - BE GLAD YOUR NOSE YOUR FACE","depth of field")

Unary terms: Two types of unary terms measure semantic (text) relatedness between the text lines and images, s_{freq} and s_{sem}, combined in a weighted sum (Eq. 3). In order to turn the similarity into a cost for the minimization, we calculate $1 - \sum weightedUnaryTerms$.

$$U(X_i) = 1 - (\lambda_1 s_{freq}(x_i) + \lambda_2 s_{sem}(x_i)) \tag{3}$$

- **Tag frequency.** s_{freq} computes the overlap between all nouns and verbs extracted from the text line and the tags associated with an image: $s_{freq} = \frac{1}{A}\sum_{a \in A} \xi_a$, with $\xi_a = \begin{cases} 1, & w_a \in \tau_{NV} \text{ occurs in } \kappa_I, \forall a \in A \\ 0, & else \end{cases}$
- **Semantic.** s_{sem} calculates the word2vec similarity between the text line words τ and the image tags κ using cosine similarity between the average representation vectors V_τ and V_κ as $s_{sem} = \text{cossim}(V_\tau, V_\kappa) = \frac{V_\tau \cdot V_\kappa}{\|V_\tau\| \|V_\kappa\|}$. This allows for similarity comparisons beyond exact word matches.

Pairwise terms: Between all possible candidate image pairs for each successive text line, a pairwise energy term is computed that is minimized to obtain a globally consistent illustration such that the storyline flows smoothly along the illustration. This pairwise potential is defined as a weighted sum (Eq. 4) of three types of consistency: style d_{sty}, color d_{col}, and content d_{cont}.

$$P(X_i, X_j) = \mu_1 d_{sty}(x_i, x_j) + \mu_2 d_{cont}(x_i, x_j) + \mu_3 d_{col}(x_i, x_j) \tag{4}$$

- **Style.** d_{sty} computes image to image coherence as the normalized Euclidean distance between the 20d style vectors of successive candidate pairs.
- **Content.** d_{cont} is obtained by the Euclidean distance between the l2 normalized CNN feature activation vectors to encourage smoothness between what successive images in our illustration depict (Fig. 7, 4).
- **Color.** d_{col} is calculated as the Euclidean distance between RGB color histograms computed between successive pairs of candidate images.

To minimize E, an NP-hard problem, we use the "sequential tree-reweighted message passing algorithm" (TRW-S) proposed by [30] whose main property is that the value of the bound is guaranteed not to decrease and, thus, at least a "local" maximum of the bound is retrieved. The weights of the parameter sets $w_U = (\lambda_1|\lambda_2)$, $w_P = (\mu_1|\mu_2|\mu_3)$ will be discussed in the following section.

6 Human Evaluation

Aligning a story with appropriate images in a pleasant style is a subjective task, especially for abstract texts such as poems and songs. Thus, in order to obtain more general ratings from a wide variety of people, we performed experiments on Amazon Mechanical Turk (AMT) to measure the quality of our method. First, we tease apart the relative contributions from various pieces of our system. We perform an experiment on the semantic connection between text and images, the unary terms, and evaluate the pairwise terms which regulate visual consistency along image sequences. Finally, we validate the quality of resulting illustrations.

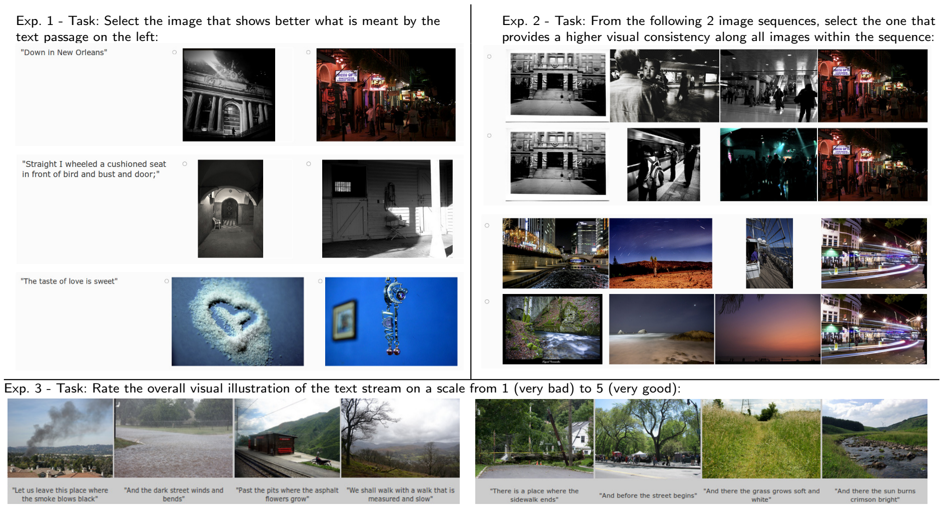
6.1 Experiment Data Set

Our experiments are designed based on the assembled data described in Sec. 4, consisting of about 200 creative texts. In total, 20 styles can be used for illustration. We randomly select 110 text-style combinations consisting of half poems and half song lyrics and ensure that every style is represented. Due to the style constraint, some text lines may result in only a few or even no image responses for a requested style. Thus, we only use text lines l with enough image candidates ($\#I_{cand_l} > 1000$) to optimize over 1000 image labels and at least one word present in the pre-trained Google word2vec representation ($\exists V_{\mathcal{T}_l}$), ensuring that we can perform a proper parameter set evaluation. Finally, to evaluate visual consistency along an image sequence, the number of image responses for each of the succeeding text lines should also be large enough. Thus, we only accept consecutive text parts T_q with M succeeding lines $T_q = \{l_{q_1}...l_{q_M}\}$ such that all consecutive lines l_{q_m} fulfill the word2vec and candidate set requirements.

6.2 User Study

People may have different internal rating systems, especially for subjective tasks. Thus, to measure relative contributions, we formulate our first experiments (Exp. 1, Exp. 2) as binary forced-choice tasks. Each pairwise preference test is designed as a data-pair selected by two parameter sets controlling different portions of the features contributing to the optimization and is presented to 5 Turkers. Depending on the type of study, data within a pair either consists of two images compared to a text line or two image sequences for evaluating visual consistency. Randomized ordering and positioning are used to negate click biases. Tasks resulting in rather unclear (2-3)-decisions out of 5 Turkers are filtered out afterwards as they are not suitable to detect trends. Based on the derived best parameter settings, we let Turkers rate the quality of final illustrations along the according text streams in a third experiment (Exp.3) on a 5pt Likert scale.

Table 2. Examples from AMT user studies. Exp. 1 (left): Evaluation of semantic text-to-image relation. Exp. 2 (right): Two example pairs of image sequences to evaluate visual consistency. Exp. 3 (bottom row): Evaluation of text stream illustrations.



Exp. 1 - Task: Select the image that shows better what is meant by the text passage on the left:

"Down in New Orleans" ○ ○

"Straight I wheeled a cushioned seat in front of bird and bust and door;" ○ ○

"The taste of love is sweet" ○ ○

Exp. 2 - Task: From the following 2 image sequences, select the one that provides a higher visual consistency along all images within the sequence:

○

○

○

○

Exp. 3 - Task: Rate the overall visual illustration of the text stream on a scale from 1 (very bad) to 5 (very good):

"Let us leave this place where the smoke blows black" "And the dark street winds and bends" "Past the pits where the asphalt flowers grow" "We shall walk with a walk that is measured and slow"

"There is a place where the sidewalk ends" "And before the street begins" "And there the grass grows soft and white" "And there the sun burns crimson bright"

Experiment 1: Text-to-image semantic. Our first experiment evaluates the semantic connection between text and images, represented by the unary terms in the optimization. We formulate our hypotheses $\mathrm{H}_{T\leftrightarrow I}$ as:

$\mathrm{H}_{T\leftrightarrow I}$:
– Both, word vectors and tag frequency are relevant in the unary terms. (tagFreq $>$ 0, wordVec $>$ 0)
– The positive influence of the word vectors is higher than of the tag frequency (wordVec $>=$ tagFreq).

The requirements described in Sec. 6.1 result in around 2110 text-image pairs. We compare binary contributions of the unary features, e.g. only wordVec $w_U = (0|1)$ against all in $(1|1)$. Table 2 (left) shows some examples of the tasks we gave to the Turkers consisting of a short text line and 2 images. Overall, including only the wordVecs has been preferred over only tagFreq (Table 3). Combining tagFreq and wordVecs has been selected over using only the one or the other feature with a 67% preference indicating that both terms are needed.

Experiment 2: Consistency along storyline. Our second experiment focuses on the visual coherence between successive images. We present pairs of sequences containing 4 images per stream to provide a reasonable evaluation set, and, without the underlying text lines to focus on the visual coherence. We evaluate the contribution of our visual features, the pairwise terms, to the optimization, formulating the hypotheses $H_{I_{seq}}$ as:

$H_{I_{seq}}$:
– All three features are necessary. (style, content, color > 0)
– Highest preference results are retrieved for relation:
color $\lesseqgtr$ content $<$ style

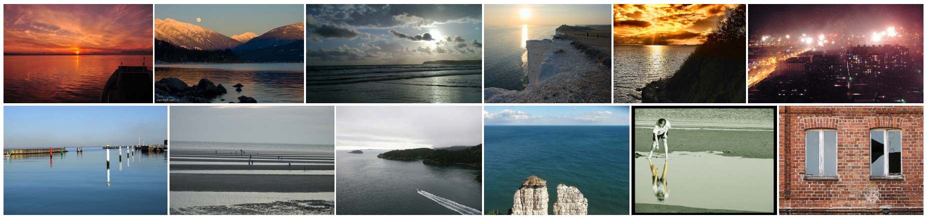
Table 3. User study results. Diffferent style groups S are identified due similar impact. S_{cont}: higher impact of content than color (e.g. "sunny", "hazy"), S_{col}: styles largely connected to color (e.g. "noir", "vintage"), S_{abst}: abstract styles (e.g. "minimal").

	Exp.1: Unary contributions			Exp.2: Pairwise contributions			
	tagFreq < wordVec	wordVec < all	tagFreq < all	col < cont	cont < sty	col < sty	noPE < allPE
S_{cont}	68%	60%	78%	85%	58%	73%	71%
S_{col}	54%	63%	60%	26%	71%	55%	74%
S_{abst}	69%	60%	80%	84%	58%	60%	47%

The constraints described in Sec. 6.1 lead to a dataset of about 1000 image sequence pairs. Based on the outcome of Exp. 1, we set $w_U = (1|1)$ and compare binary contributions between the pairwise feature terms to retrieve relations between them. Table 2 (right) shows some tested sequences. Results are shown in Table 3. Style was always preferred over the other features to ensure coherence. In this experiment, we identified different groups S of styles. Styles like "sunny" profit from a higher contribution of content than color (S_{cont}, 85% preference). Other styles, e.g. "noir" largely depend on color being preferred 74% over content (S_{col}). Rather abstract styles like "minimal" are not as suitable for auto-illustration as (0|0|0) was preferred 53% over (1|1|1) (S_{abst}).

The results of the study demonstrate the importance of distinguishing between certain styles. Thus, based on the performed binary experiments and the obtained relations, we experimentally obtained different parameter sets relating the proportions of the features for different groups of styles and, similar to experiment 2, tested them against all weights set to 1. Most of the styles worked best for a weight set of $w_U = (.8|1), w_P = (1|.5|.2)$, e.g. "hazy" 80%. Contrarily, for "horror" (1|1|1) was preferred in average 92% over partial combinations. However, very color depending styles worked better combining the features with $w_U = (.8|1), w_P = (1|.2|.5)$, thus setting color > content, e.g. "noir" was preferred 90% over all set to 1 and "vintage" 80%.

Experiment 3: Text illustration. Based on the previously derived parameter settings, we let Turkers rate the quality on a subset of 45 illustrations along text lines on a 5pt Likert scale from 1 (very bad) to 5 (very good). The subjective outcome of our system makes it challenging to obtain scores rating the overall quality. However, for many styles the resulting mean μ was around 4 indicating good quality, e.g., "long exposure" $\mu = 4.0$ ($\sigma = .91$), "noir" $\mu = 3.9$ ($\sigma = 1.07$). We tended to find that results in the style group G_{cont} had stronger decisions



The sea is calm tonight.

The tide is full, the moon lies fair

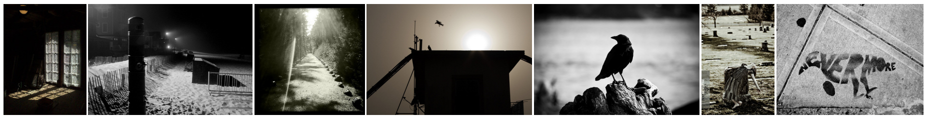
Upon the straits; on the French coast the light

Gleams and is gone; the cliffs of England stand,

Glimmering and vast, out in the tranquil bay.

Come to the window, sweet is the night-air!

Fig. 4. Poem "M. Arnold - DOVER BEACH" in: "sunny" (top), "minimal" (bottom).



"Surely," said I, "surely that is something at my window lattice"

Ghastly grim and ancient raven wandering from the Nightly shore

Respite - respite and nepenthe, from thy memories of Lenore!

What this grim, ungainly, ghastly, gaunt and ominous bird of yore

Meant in croaking "Nevermore."

It shall clasp a sainted maiden whom the angels name Lenore

Quoth the Raven, "Nevermore."

Fig. 5. Poem "E. A. Poe - THE RAVEN" illustrated in style "noir". Note "nevermore" presented in form of a raven and the different selections for "croaking" and "quoting".

(smaller σ-values). "Minimal" only obtained a top-two box acceptance of 37%. However, non-abstract styles obtained top-two box acceptance rates between 70% and 80%, indicating high acceptance of our results, e.g., "sunny" 75%.

7 Results and Discussion

The main challenge of our approach is to balance semantic relevance with producing an illustration that both depicts the requested style and demonstrates strong visual coherence along the illustration. Fig. 5 provides a consistent visual appearance of the style "noir" while preserving the meaning of the underlying text lines, even distinguishing between the raven "croaking" and "quoting".

The weak nature of tags and polysemy makes this problem highly challenging, e.g., in Fig. 6, all images are tagged with "cloud" although the last image does not show a cloud. However, sometimes our method works surprisingly well, e.g., in Table 2 (left) the text "The taste of love is sweet" shows an amazing result with our features, providing an idea of taste. Overall, the the nice repetitive structure of creative texts allows us to search an image for each text line instead of forcing text splits that can lead to wrongly combined words. Additionally, restricting the candidates per text line by the style prediction attribute makes it even more challenging to create a semantically relevant illustration. In Fig. 1, the line "Well, I get down on my knees and pretend to pray" works nicely for the first and last row with the styles "hdr" and "serene", but provides a rather strange X-ray image of a knee for "noir" as there was no better candidate available. Overall, as shown in Sec. 6.2, the way we combine our textual features generally supports selecting suitable images illustrating the meaning of the text.

Further, recurring lines in poems as well as refrains in songs often serve as stylistic device to strengthen the main message by repetition. Thus, if similar

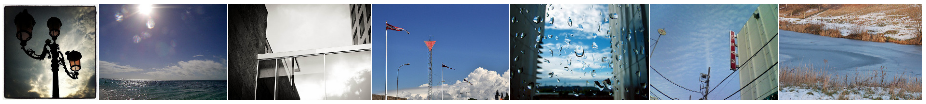
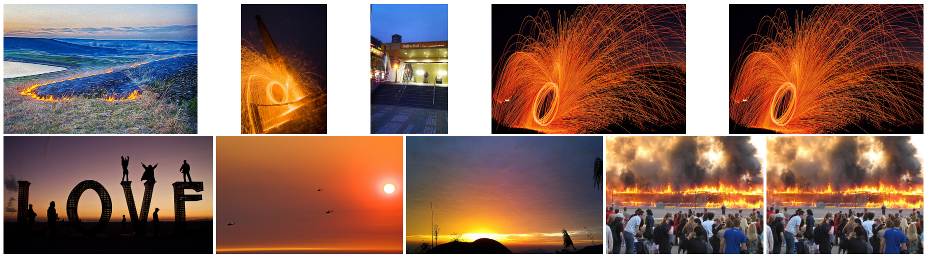


Fig. 6. Poem "W. - I WANDERED LONELY CLOUD". The text line "I wandered lonely as a cloud" is presented in different illustration styles (horror, sunny, noir, bright, depth of field, geom. composition, texture). All image tags include the word "cloud".


Love is a burning thing

And it makes a fiery ring

Bound by wild desire

I fell into a ring of fire

I fell into a burning ring of fire

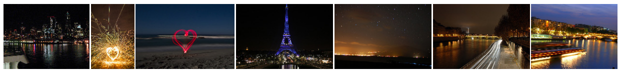
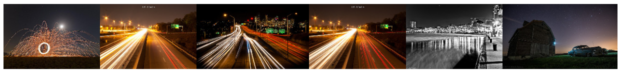
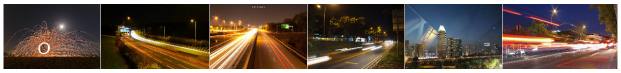
Fig. 7. Song "J. Cash - RING OF FIRE" illustrated with the styles "long exposure" (top) and "sunny"(bottom). Recurrence is provided for similar texts parts.

content is described in text lines, the images should provide similar or even identical visualizations, e.g., Fig. 7 visualizes recurring images for lines with similar descriptions. Often, such creative texts deal with a main theme. We capture its global idea by first providing pre-selection of similar candidate lists by recurrence, and, then, the content feature selects images with highest content similarity along the storyline. Table 4 shows an example of candidate lists sorted by highest style prediction (right) along succeeding lines (left) and estimated optimization results (middle). We can observe that the images are selected properly with high coherence in their visual appearance and style along the sequence.

Overall, not every style is similarly suitable to illustrate text. In our user study (Sec. 6.2) we identified different style groups and retrieved lowest acceptance rate for abstract styles. However, even for such styles we were able to retrieve nice results, e.g., in Fig. 4 for the abstract style "minimal". Additional examples for different styles are shown in Fig. 7 or 9. Please see the supplemental material for longer and more extensive illustration results.

Finally, rather than restricting our framework to creative texts only, we enable the input of new text data of arbitrary type and length. For a selected style, the text is then parsed and illustrated visually consistent by our system with one image per text line out of the YFCC14M images.

Table 4. Example for candidate lists along part of song "The doors - LIGHT MY FIRE" in illustration style "long exposure". Left: succeeding text lines. Second column: selected optimization result. Right: candidate images per text line.

Try now we can only lose		
And our love become a funeral pyre		
Come on baby light my fire		
Try to set the night on, fire yeah		

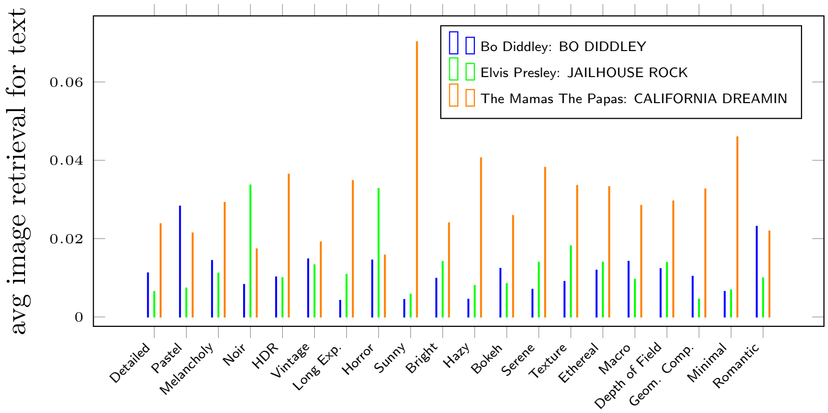


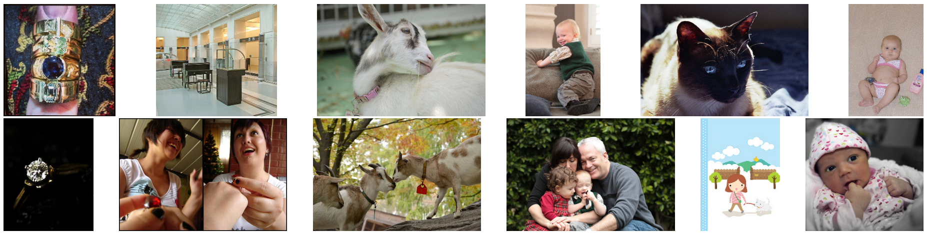
Fig. 8. Examples for average style-image-responses for some song lyrics for all 20 styles. Highest peaks indicate main moods of the text.

8 Applications

Finally, our system can enable applications such as identifying the style of a text using our candidate selection process or automatically generating a music video by illustrating complete songs in different styles.

Text Style from Candidate Selection. Restricting the number of retrieved images for text lines by the style constraint $I_{sty} > 50\%$ (Sec. 5.1) leads to interesting insights about the connection between text and styles. We calculated the average number of retrieved images for the text lines in a story relative to the available amount of images for a certain style $\#I_{sty}$ in the YFCC14M subset. Thus, for a story text T and a style sty, the average style-image-response T_{sty} is calculated as shown in Eq. 5. Note, that only text lines $l \in L$ with number of candidate images $\#I_{cand_l} > 0$ are considered.

$$T_{sty} = I_{ret}(sty) = \frac{\sum_{l \in L} I_{cand_l}(\text{sty})}{\#l} \cdot \frac{1}{\#I_{sty}} \;, \quad \text{with } \forall l \in L, \#I_{cand_l} > 0 \tag{5}$$



Bo Diddley bought his babe a diamond ring

He'd better not take the ring from me

Bo Diddley caught a nanny goat

To make his pretty baby a Sunday coat

Bo Diddley caught a bear cat

To make his pretty baby a Sunday hat

Fig. 9. Song "B. Diddley: BO DIDDLEY". Highest average style-image-responses are obtained for the styles "pastel" (top) and "romantic" (bottom).

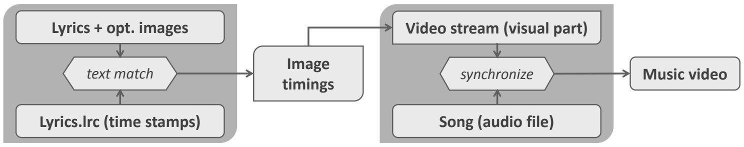


Fig. 10. Music video generation pipeline.

Fig. 8 demonstrates this idea, showing the resulting average style-image-response of stories for all provided styles. Some texts have a peak in one or two styles, e.g., "The Mamas The Papas: CALIFORNIA DREAMIN" in "sunny". Others even have peaks in connected styles, e.g., "E. Presley - JAILHOUSE ROCK" in "noir"+"horror" or "B. Diddley: BO DIDDLEY" (Fig. 9) in "pastel" + "romantic". Interestingly, these styles do seem to indicate the main moods of the texts as they often work better as styles with a lower image-style-response.

Music Video Generation. Further, we enable the generation of simple music videos. An overview is given in Fig. 10. For that purpose, our system outputs a file listing the song lyrics and selected images links. Additionally, the audio version of the song and its ".lrc"-file are needed. The LRC format [31], is usually used in karaoke to align song lyrics with the music. The structure simply consists of the text lines of a song and its associated time-stamps. Optionally, additional ID tags indicating artist and song meta information might be attached.

For the music video generation, we start by matching the text lines between the ".lrc"-file and our "lyrics2images"-file. To be robust against spelling errors, we compare the stems of the words text line to obtain the time-stamps for the images. Our tool converts them into duration timings for each image. A video stream is then simply generated by displaying the images in their proposed ordering and duration timings similar to a slide show. This video stream is joined together with the audio file resulting in a music video. The synchronization is already provided by the image timings. However, beat detection could improve synchronization in future work.

9 Conclusion

Given a preferred style the presented pipeline automatically generates an illustration for a poem or song. Our framework optimizes both semantic relevance and visual coherence while selecting images, exploiting recent advances in convolutional neural networks for image and style classification. The generated sequences have been evaluated in a user study, indicating that combining multiple features improves over simpler image selection processes and obtaining highest acceptance rates for non-abstract styles. We also demonstrate applications to story style classification and music video generation.

Acknowledgement. This work originated from a research stay of Katharina Schwarz at the University of North Carolina (UNC).

References

1. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. In: NIPS. (2013) 3111–3119
2. Karayev, S., Trentacoste, M., Han, H., Agarwala, A., Darrell, T., Hertzmann, A., Winnemoeller, H.: Recognizing Image Style. In: BMVC. (2014)
3. Snavely, K.N.: Scene Reconstruction and Visualization from Internet Photo Collections. PhD thesis, University of Washington (2009)
4. Frahm, J.M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.H., Dunn, E., Clipp, B., Lazebnik, S., Pollefeys, M.: Building Rome on a Cloudless Day. In: ECCV. (2010) 368–381
5. Hays, J., Efros, A.A.: Scene Completion Using Millions of Photographs. In: SIGGRAPH, ACM (2007)
6. Averbuch-Elor, H., Wang, Y., Qian, Y., Gong, M., Kopf, J., Zhang, H., Cohen-Or, D.: Distilled Collections from Textual Image Queries. Computer Graphics Forum (2015) 131–142
7. Kim, G., Xing, E.P.: Reconstructing Storyline Graphs for Image Recommendation from Web Community Photos. In: CVPR. (2014) 3882–3889
8. Kim, G., Sigal, L., Xing, E.P.: Joint Summarization of Large-scale Collections of Web Images and Videos for Storyline Reconstruction. In: CVPR. (2014) 4225–4232
9. Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A.C., Berg, T.L.: Baby talk: Understanding and generating image descriptions. In: CVPR. (2011) 1601–1608
10. Ordonez, V., Kulkarni, G., Berg, T.L.: Im2Text: Describing Images Using 1 Million Captioned Photographs. In: NIPS. (2011) 1143–1151
11. Karpathy, A., Li, F.F.: Deep visual-semantic alignments for generating image descriptions. In: CVPR. (2015) 3128–3137
12. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: CVPR. (2015) 3156–3164
13. Devlin, J., Cheng, H., Fang, H., Gupta, S., Deng, L., He, X., Zweig, G., Mitchell, M.: Language Models for Image Captioning: The Quirks and What Works. In: ACL. (2015) 100–105
14. Fang, H., Gupta, S., Iandola, F., Srivastava, R.K., Deng, L., Dollar, P., Gao, J., He, X., Mitchell, M., Platt, J.C., Zitnick, C.L., Zweig, G.: From Captions to Visual Concepts and Back. In: CVPR. (2015) 1473–1482
15. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A.C., Salakhutdinov, R., Zemel, R.S., Bengio, Y.: Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In: ICML. (2015) 2048–2057
16. Coyne, B., Sproat, R.: WordsEye: An Automatic Text-to-scene Conversion System. In: SIGGRAPH, ACM (2001) 487–496
17. Spika, C., Schwarz, K., Dammertz, H., Lensch, H.P.A.: AVDT - Automatic Visualization of Descriptive Texts. In: VMV. (2011) 129–136
18. Zitnick, C.L., Parikh, D.: Bringing Semantics into Focus Using Visual Abstraction. In: CVPR. (2013) 3009–3016
19. Zitnick, C.L., Parikh, D., Vanderwende, L.: Learning the Visual Interpretation of Sentences. In: ICCV. (2013) 1681–1688
20. Kong, C., Lin, D., Bansal, M., Urtasun, R., Fidler, S.: What Are You Talking About? Text-to-Image Coreference. In: CVPR. (2014) 3558–3565

21. Joshi, D., Wang, J.Z., Li, J.: The Story Picturing Engine—a System for Automatic Text Illustration. TOMCCAP (2006) 68–89
22. Schwarz, K., Rojtberg, P., Caspar, J., Gurevych, I., Goesele, M., Lensch, H.P.A.: Text-to-Video: Story Illustration from Online Photo Collections. In: KES. (2010) 402–409
23. Kim, G., Moon, S., Sigal, L.: Ranking and Retrieval of Image Sequences from Multiple Paragraph Queries. In: CVPR. (2015) 1993–2001
24. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python. 1st edn. O'Reilly Media, Inc. (2009)
25. Fellbaum, C., ed.: WordNet: an electronic lexical database. MIT Press (1998)
26. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. CoRR (2013)
27. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR (2014)
28. Thomee, B.: Yahoo! Webscope dataset YFCC-100M. http://labs.yahoo.com/Academic_Relations (2014)
29. Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.: The New Data and New Challenges in Multimedia Research. CoRR (2015)
30. Kolmogorov, V.: Convergent Tree-Reweighted Message Passing for Energy Minimization. IEEE Trans. Pattern Anal. Mach. Intell. (2006) 1568–1583
31. Shiang-shiang, K.D.: Information about LRC. http://www.mobile-mir.com/en/HowToLRC.php (2012)