Diffusion Models as Data Mining Tools

Ioannis Siglidis[1], Aleksander Holynski[2], Alexei A. Efros[2], Mathieu Aubry[1], and Shiry Ginosar[2]

[1] LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, Marne-la-Valle, France
[2] University of California, Berkeley

https://diff-mining.github.io/

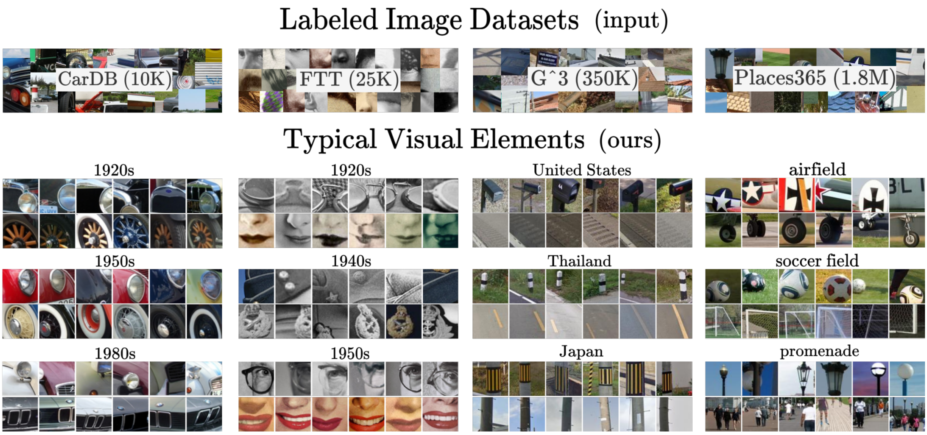


Fig. 1: Mining typical visual elements with diffusion models. We demonstrate how to use diffusion models to mine visual data through a simple pixel-based score and a standard clustering approach. We present high-quality mining results for a diverse range of datasets (from left to right: 10,130 photographs of cars tagged with a creation year between 1920-1999 [24], 24,874 portraits from the 19th to the 21st century [9], 344,224 Street View images tagged with country names [28], and 1,803,460 images of scenes images associated with descriptive names [49]). Our results highlight both expected elements and more unforeseen ones.

Abstract. This paper demonstrates how to use generative models trained for image synthesis as tools for visual data mining. Our insight is that since contemporary generative models learn an accurate representation of their training data, we can use them to summarize the data by mining for visual patterns. Concretely, we show that after finetuning conditional diffusion models to synthesize images from a specific dataset, we can use these models to define a typicality measure on that dataset. This measure assesses how typical visual elements are for different data labels, such as geographic location, time stamps, semantic labels, or even the presence of a disease. This analysis-by-synthesis approach to data mining has two key advantages. First, it scales much better than traditional correspondence-based approaches since it does not require explicitly comparing all pairs of

visual elements. Second, while most previous works on visual data mining focus on a single dataset, our approach works on diverse datasets in terms of content and scale, including a historical car dataset, a historical face dataset, a large worldwide street-view dataset, and an even larger scene dataset. Furthermore, our approach allows for translating visual elements across class labels and analyzing consistent changes. Project page: `https://diff-mining.github.io/`.

Keywords: Visual Data Mining · Diffusion Models

1 Introduction

Visual data mining aims to discover patterns within large visual corpora such as collections of street view panoramas [12, 23], historical images of faces [9, 13] or photographs of cars [10, 24]. This paper proposes a novel idea: to turn generative models trained for image synthesis into a scalable method for visually mining image datasets. Generative models digest massive amounts of data, which they implicitly store in their weights. Our central insight is that we can use this learned summary of the visual input to identify the most typical image regions. This unconventional use of a diffusion model for studying its training data demonstrates that generative models are potent tools beyond synthesis—for data mining, summary, and understanding.

Our target task, mining for informative visual patterns, is challenging. Unlike text, where words act as discrete tokens that we can directly compare, the visual world seldom contains exactly repeating elements. Even common simple visual elements, such as windows, can have different colors and different numbers of panes; they may be seen from various viewpoints, and they may be located at multiple positions as part of different facades. The standard approach to visual data mining [12, 24, 40] involves learning data-specific similarities with relevant invariances (*e.g.*, such that different-looking windows will be similar) and using them to search for discriminative patterns. However, these techniques are not easily scalable since one must apply them across all pairs of visual elements within all pairs of images in the dataset. The similarity graph between visual elements scales quadratically with the size of the dataset. In contrast, our proposed analysis-by-synthesis approach does not require pairwise comparisons between different visual elements and thus scales to very large datasets.

The approach we propose takes as input a dataset with image-level tags, such as time [9, 24], geography [28], or scene labels [49]. Our goal is to provide a visual summary of the elements typical of the different tags, such as the common elements that enable us to determine the location of a streetview panorama. To arrive at this summary, we first finetune a conditional diffusion model on the target dataset. We then use the finetuned model to define a pixel-wise typicality measure by assessing the degree to which the label conditioning impacts the model's reconstruction of an image. We mine visual elements by aggregating typicality on patches, selecting the most typical ones, and clustering them using features extracted from the finetuned model [44]. As visualized in Fig. 1, this

leads to clusters of typical visual elements that summarize the most characteristic patterns associated with the tags available in the input dataset. For example, our face results highlight iconic elements, such as aviator glasses in the 1920s and military hats in the 1940s, and more subtle details, such as period-typical glasses or make-up. Interestingly, our results on street-view data highlight details that are similar to the ones presented in geographical understanding websites [1, 3, 4], popularized through the GeoGuessr game [2], such as typical parts of utility poles, bollards, or architecture. To our knowledge, no existing visual mining method has demonstrated such high-quality results on diverse datasets.

2 Related Work

Visual data mining. Visual data mining turned the manual and subjective process of comparing photographs (*e.g.*, [22]) into algorithmic methods for summarizing image data, such as architectural details [12, 23], fashion [9, 13, 30], industrial design [17], and art [19, 39, 41] by locating visual patterns. This has mainly been achieved using discriminative techniques such as clustering or contrastive learning. For example, [24] demonstrated how correspondence based mining across time can be achieved in a dataset of objects of similar parts, namely cars, and [12] showed that geographically representative image elements can be automatically discovered from Google Street View imagery in a discriminative manner. However, such traditional data mining approaches do not scale to large modern datasets. Indeed, they require pairwise comparisons between all the visual elements of each image to the entire dataset in order to locate nearest neighbors and establish clusters. Notably, the discriminative clustering algorithm of [12] requires training a separate linear SVM detector for each visual element- a computationally prohibitive approach when considering multiple possible visual elements for the purposes of analysis. In contrast, our approach is scalable to very large datasets. Closer to our work, generative model have been trained to analyze the evolution of faces [9] and cars [10] across time. However, these two works essentially perform image translation, and do not enable actual mining of typical elements in the datasets.

Diffusion models. Diffusion models have gained popularity in recent years due to their stability in training and effectiveness in modeling complex multimodal distributions [11, 15, 16, 20, 42]. These models are capable of generating high-quality imagery conditioned on input signals beyond categorical labels, like text [35, 36, 38], and can further incorporate additional modalities [26, 48]. In addition to generating images from scratch, diffusion models have been used extensively for instruction-driven image-to-image translation [7, 14, 32, 33, 45]. It has also been shown that pre-trained text-to-image diffusion models encode strong priors for natural scenes, allowing their internal features to be used for secondary tasks [29, 44, 47]. They can easily be adapted for new tasks or to new data distributions through minimal finetuning [7, 37, 48].

Beyond mere image synthesis, generative image models, and in particular diffusion models, have been used as data augmentation engines. While most

machine learning approaches treat the data as fixed and improve the learning algorithm, works such as [6, 8, 18] fix the learning algorithm and augment the training data, using generative models to synthesize large amounts of synthetic data. In contrast, we present a new way to use generative models, aiming to gain insights about their training data.

3 Data Mining via Diffusion Models

Our approach turns generative models into data mining tools. It relies on finetuning a conditional stable diffusion model trained for image synthesis, using it to extract a summary of the visual world. We start by reviewing diffusion models and the techniques we leverage in section 3.1. In section 3.2, we introduce our measure of typicality, which allows us to measure how the class label conditioning affects the synthesis of an image by the diffusion model. In section 3.3, we describe how we aggregate typicality on patches to mine typical visual elements and cluster them to summarize the training data.

3.1 Preliminary

Diffusion models. Diffusion models are generative models trained to transform random noise $\epsilon \sim N(0, 1)$ into a target distribution $\mathcal{X}$. The denoising process is iterative, indexed by a step index t. A diffusion model $\epsilon_\theta(z, t)$, with parameters θ, takes as input an image z to be denoised at the fractional timestep t.

During training, a training sample x is artificially noised at a strength associated to a uniformly sampled step t, by mixing the image with a randomly sampled Gaussian noise image ϵ in what is known as the *forward process*:

$$\text{noise}(x, \epsilon, t) = \sqrt{\bar{a}_t}x + (1 - \sqrt{\bar{a}_t})\epsilon, \tag{1}$$

where $\sqrt{\bar{a}_t}$ defines the noise mixing coefficient which varies over the denoising process. The learnable denoising model ϵ_θ takes as input both a noised image and the corresponding noising step and is trained to predict the noise image (or equivalently the denoised image) using a loss:

$$L_t(x, \epsilon) = ||\epsilon_\theta(\text{noise}(x, \epsilon, t),\ t) - \epsilon||^2. \tag{2}$$

At test time, the target distribution can be sampled by transforming the noise distribution through an iterative denoising process [15, 43], in which a randomly sampled image of noise z_T is gradually denoised according to the model's predictions.

Conditional Diffusion Models. Diffusion models can be extended to take a conditioning c associated with the image content as an additional input. This leads to a model $\epsilon_\theta(z, t, c)$ that depends on the noisy image z, the time step t,

the conditioning c, and a loss $L_t(x, \epsilon, c)$. In our case, c will correspond to the CLIP [34] text features of the class label associated with the image.

Latent diffusion models. We use a variant of diffusion models known as a *latent* diffusion model (LDM) [36]. Instead of modeling the source data distribution, LDMs model its distribution in the latent space of a variational autoencoder [21]. Working in the latent space reduces the complexity of the data distribution. It thus significantly reduces both the number of parameters of the diffusion model and the amount of training samples necessary to learn a good model.

3.2 Typicality

We design our measure of typicality based on the following intuition: a visual element is typical of a conditioning class label (*e.g.*, country name or date) if the diffusion model is better at denoising the input image in the presence of the label than in its absence. We, therefore, design typicality as a ranking measure across pixels between the ground-truth conditioning c and the null conditioning $\varnothing$. We define the *typicality* of an image x given the class label conditioning c as:

$$\mathbf{T}(x|c) = \mathbb{E}_{\epsilon,t}[L_t(x, \epsilon, \varnothing) - L_t(x, \epsilon, c)], \tag{3}$$

where t is sampled uniformly from $[0, 1]$, and ϵ is sampled according to the noise distribution $N(0, 1)$. This typicality measure enables us to sort visual elements from a specific class by how typical they are of that class (see supplementary material for additional formal motivation of this measure). Our typicality measure is related to the image-level classification approach of Li *et al.* [25], but it is built for pixel-based analysis and data mining. Unlike Li *et al.*, we find that reducing the sampled range of t to $[0.1, 0.7]$ improves the quality of our results, as the tails can contribute uninformative yet typical samples, as we show in the supplementary material.

3.3 Mining for Typical Visual Elements

Conditioning and finetuning. To mine typical visual elements for a given class, we use the text class label conditioning c in the form of its CLIP text features [34]. We convert the tags associated with the datasets to text using the embeddings of the following sentences: "A car/portrait from the {decade}s." for faces and cars ("A car/portrait." for the null conditioning $\varnothing$), "A Google streetview image of {country}." for streetview data ("A Google streetview image." for the null conditioning $\varnothing$), and "An image of {scene}." for images of the Places dataset [49] (empty string for the null conditioning $\varnothing$). We finetune a latent diffusion model [36] on the target dataset by optimizing the reconstruction loss (Equation 2) given the conditioning. We use Stable Diffusion V1.5 [36] as a base model in all our experiments.

Patch-based analysis. To find condition-specific visual elements, we compute our typicality scores over patches of images by averaging typicality in the area

of a patch[3]. To identify the set of most typical visual elements for a dataset we pick the 5 most typical non-overlapping patches in each image according to the patch typicality, and select the 1000 most typical patches over all the dataset.

Clustering visual elements. We cluster the most typical patches using k-means [27] with 32 clusters. To cluster elements, we embed them with DIFT [44] features, computed at timestep $t = 0.161$ using our finetuned models. For visualization, we rank clusters by the median typicality of their elements in decreasing order and the elements within a cluster by the distance to the centroid in increasing order.

4 Experiments

We showcase the effectiveness of our approach in summarizing visual data for a wide variety of datasets. First, in Section 4.1, we introduce the datasets used in our experiments. Second, in section 4.2, we evaluate the ranking given by our typicality measure. Third, in Section 4.3, we discuss our main result, the mined visual summaries of the analyzed datasets, and compare with Doersch *et al.* [12]. Finally, we discuss the limitations of our approach in Section 4.4.

4.1 Datasets

We experiment with four diverse datasets. CarDB [24] and FTT [9] have already been used for visual mining and include a few tens of thousands of images. G^3 [28] and Places [49] are much larger with 344K and 1.8M images respectively, and, to our knowledge, have never been used for visual mining.

Cars. The CarDB dataset [24] contains 10,130 photos of cars from 1920 to 1999, collected from `cardatabase.net`. They are labeled with creation years, which we bin into decades for our analysis. This dataset contains cars seen from various viewpoints and in diverse environments. As a result, extracting time-informative elements is challenging. We rescale all images to a height of 256 pixels while preserving their original aspect ratio.

Faces. The Faces Through Time (FTT) Dataset [9] contains 24,874 images of notable people from the 19th to 21st century, with roughly 1,900 images per decade, sourced from Wikimedia Commons. All photos are of size 256x256 pixels.

Geo. The G^3 [28] dataset contains images obtained from crops of street-view panoramas, diversely sampled worldwide, of which we selected 344,224 images, which we rescaled to 512x756 pixels. This dataset is challenging because of the small details that characterize a scene's appearance and scale. We focus on the 8 countries with the largest number of panoramas (United States, Japan, France, Italy, United Kingdom, Brazil, Russia, and Thailand) and two countries with

[3] Since we use latent diffusion, the loss for arbitrary patches requires upsampling the feature maps to the original image resolution.

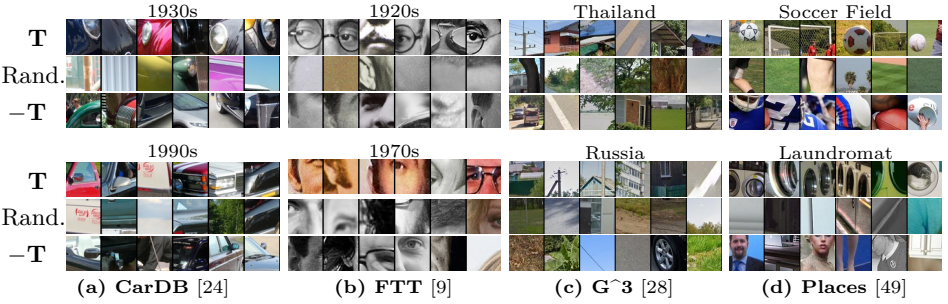


Fig. 2: Typical elements are informative of the conditioning label. We visualize the top-6 patches ranked according to typicality (**T**) with respect to the conditioning class label, negative typicality (−**T**), and randomly (Rand.). The two rows correspond to different classes from each of the four datasets.

fewer images (Nigeria and India). We finetune the network using all images from these countries, but we only mine a random subset of 1000 images.

Places. The high-resolution version of the Places dataset [49] contains 1,803,460 million images from 365 place categories associated with their labels, with a minimum dimension of 512 pixels. For mining we only use the validation dataset, which contains 100 images per scene category.

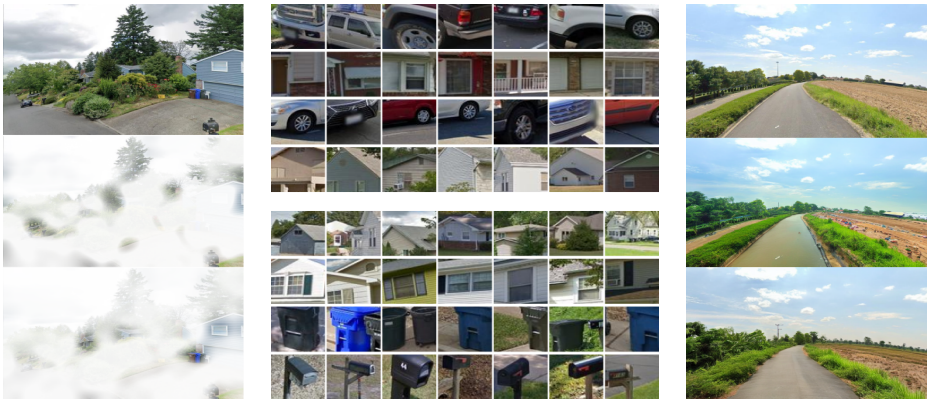
4.2 Typicality Measure Evaluation

Typicality score for patches. Fig. 2 shows the most and least typical patches according to our typicality measure and random patches for the four datasets. We note that the most typical patches are unique to each class and more discriminative than random patches, while the least typical patches are uninformative of the conditioning label.

Effect of finetuning. Unsurprisingly, we found that finetuning the diffusion model on the dataset of interest was critical to the quality of our results. First, on a given image, finetuning changes the spatial distribution of typicality, prioritizing elements more correlated with the training labels (see Fig. 3a). Second, in Fig. 3b, we show the most typical clusters identified before and after finetuning. The patches selected after finetuning avoid the biases in the training data of the base model and are more specific to the G ^3 dataset, identifying elements such as post-boxes. We also demonstrate this quantitatively in Section 5.2 for our application to X-ray images. Third, finetuning enables better translation between labels (see Sec. 5.1), as can be seen in Figure 3c, allowing vegetation, roads, road tracks, and utility poles to be translated consistently across the class labels in the parallel dataset (which can be found in the supplementary material).

4.3 Clusters of Typical Visual Elements

In this section, we analyze our visual summary of each dataset, obtained by clustering the typical visual elements for the different class labels. We show our



(a) Typicallity **(b)** Clusters **(c)** Parallel Translation

Fig. 3: Effect of finetuning. **(a)** For the same USA image (top), finetuning changes the spatial allocation of typicality before (middle) and after (bottom) finetuning. **(b)** This results in different typical clusters (USA), which, after finetuning (bottom), select for more typical elements like mailboxes. **(c)** Translation (Sec. 5.1) of a picture of a road from France (top) to Thailand without finetuning (middle) suffers from data biases in the base model turning the road into a river and erasing utility poles. After finetuning on the G^3 dataset (bottom), the translated image is more consistent with the original.

summaries for Cars, Faces, Geo, and Scenes in Figs. 4, 5, 6, 7 respectively. For all datasets, we show the top-6 typical elements of the top-6 clusters ranked by the median typicality of their elements for selected class labels. We analyze the resulting clusters for each dataset inline in the figure captions for ease of viewing. Our complete clusters can be found in the supplementary material.

Comparison to Doersch *et al.* [12]. As the Matlab implementation of [12] is obsolete and hardware-specific, we reimplement their method in Python and release this reimplementation with our code. In Fig. 8, we show the results of this approach when applied to the same subset of the G^3 dataset as our approach. Similar to the original paper, we rank the trained detectors by *discriminativeness*, *i.e.*, the percentage of the top-50 final matches inside the positive set [12], and for each we show its top 6 matches. The results produced with the Doersch *et al.* method demonstrate more textures, appear much less semantic, and contain much more similar elements than ours. Note that the results in the original Doersch *et al.* paper do not show such failures, and in particular much less vegetation, simply because the paper used a curated and non-publicly available dataset of images focused on selected cities extracted from Google Street View.

4.4 Limitations

Although our method makes the first step towards utilizing generative models for data mining, it comes with limitations. We visualize our two main failure modes in Fig. 9. First, clustering elements using k-means can lead to mixed clusters containing different categories of samples (Fig. 9a) or produce repetitively similar



(a) 1920s (b) 1930s (c) 1980s

Fig. 4: Clusters of CarDB [24] visual elements. Our visual summaries of typical car elements show elements unique to a period and elements that evolve with time. Evolving elements include the shapes of the car's body or headlights, which are parts of the 6 most typical clusters for most periods. More specific elements include running boards in the 1920s (**(a)**, 6th row) or large engine side grills in the 1930s (**(b)**, 3rd, 4th and 6th row). In the 1980s **(c)**, we observe two typical yet very discrete clusters of car design styles, of the curvy French 2CV (1-4 row) juxtaposed to the square American *chevy*-style cars (5-6 rows).

clusters. Second, our method identified data artifacts (Fig. 9b) that are related to noisy printing or scanning of old photographs or post-processing artifacts of street view images, which are highly typical but irrelevant to our purpose. Interestingly, in the case of street-view data such artifacts are suggested in GeoGuessr [2] advice websites [1, 3, 4], as shortcuts for geolocation.

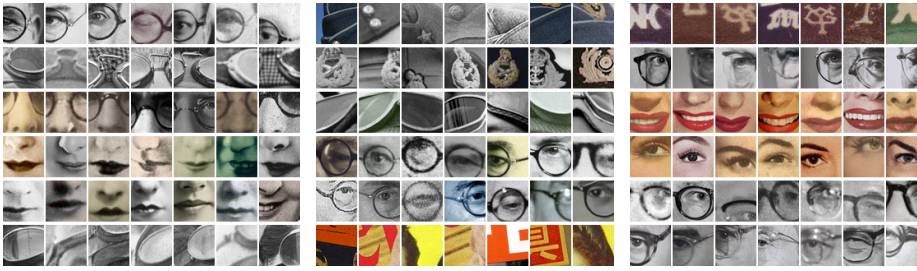
5 Applications

Our typicality score allows us to explore two different applications. First, in Section 5.1, we translate geographical elements across locations and mine typical translations. Then, in Section 5.2, we show how disease localization emerges from typicality when training to generate frontal X-rays of patients, of various diseases.

5.1 Analyzing Trends of Visual Elements

Having a diffusion model finetuned on a dataset of interest enables further applications that were not possible with previous visual mining approaches [9, 12, 13, 24]. One new application is the summary of variation of typical visual elements across different class labels. As a case study, we use the G^3 dataset to discover and summarize how *co-typical* elements, such as windows, roofs, or license plates, vary across locations. We start by using our finetuned diffusion model to create a "parallel dataset", by translating all the images in our mining dataset to all locations, then define a co-typicality measure.

Generating a parallel dataset. We first use Plug and Play [45] to translate input images from one location to another, which we denote by $x^{c_0 \to c}$, where c_0 is the initial country and c is the target country. We translate 1000 images for



(a) 1920s (b) 1940s (c) 1950s

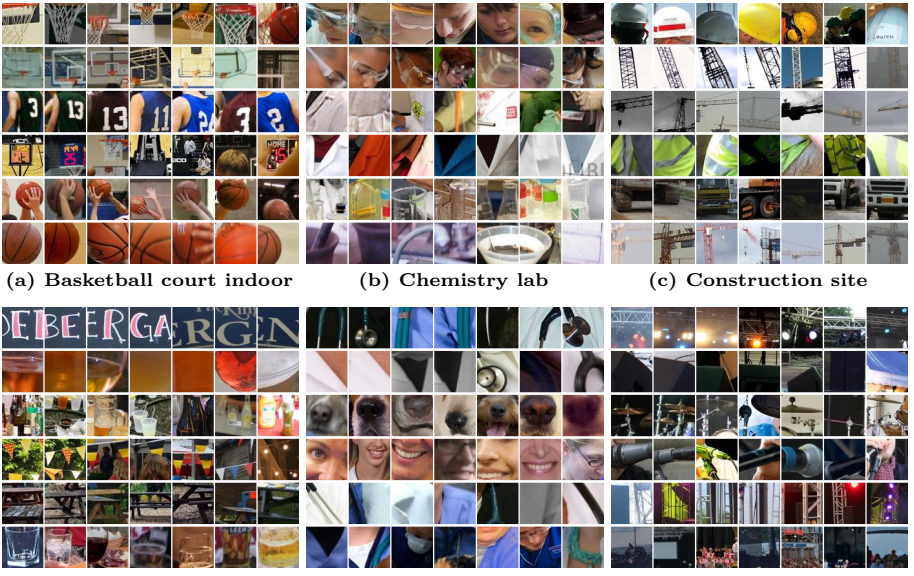
Fig. 5: Clusters of FTT [9] visual elements. Our cluster analysis of faces revealed that eyeglasses of varying designs are indicative of a portrait's decade throughout the history captured by FTT. Observing the 6 most typical clusters for the 1920s **(a)**, the 1940s **(b)**, and the 1950s **(c)**, we see how the shape of glasses is highly informative of each period. We also located fashion items that uniquely trended only in a particular period, such as aviator goggles in the 1920s (2nd row), military caps in the 1940s (1st and 2nd row), and baseball caps in the 1950s (1st row). Consistent with prior analysis [13], we also found clusters corresponding to smiles and makeup.



(a) United States (b) Russia (c) Brazil

(d) France (e) Japan (f) Thailand

Fig. 6: Clusters of G^3 [28] visual elements. Our geographic clusters show a wide diversity of typical elements across different countries. We found architectural elements such as roofs, facades, or windows among the most typical elements in all countries. For example, **(a)** the "double hung" American windows (2nd row), **(d)** French roof windows (1st-4th row), or **(f)** covered pathways in Thailand (4th row). Utility poles are ranked second in Russia and Thailand and 5th in Brazil. We also found typical objects that are unique to a single country, such as **(a)** American garbage cans and post boxes (3rd, 4th row), **(c)** protective guard rails in Brazil (2nd row), **(e)** Japanese electricity warning signs and exterior wall tiles (1st, 2nd row), and **(f)** Thai Bollards (1st row).



(a) Basketball court indoor (b) Chemistry lab (c) Construction site

(d) Beer garden (e) Veterinarians office (f) Stage outdoor

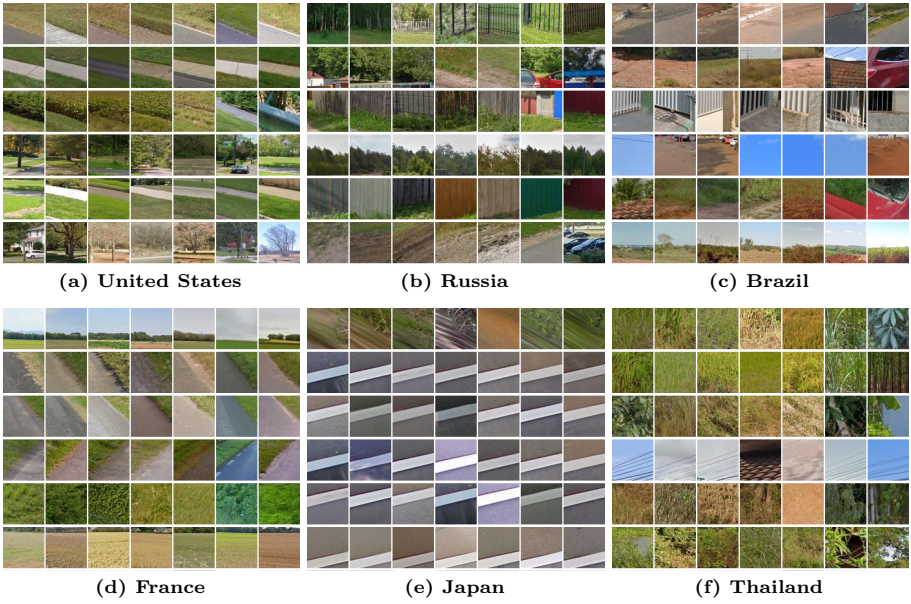
Fig. 7: Clusters of Places365 [49] visual elements. Unlike the other datasets we analyze, each class label correlates with objects of different categories in the scenes dataset, as different scenes contain objects of different categories. Yet, our approach can still summarize a large variety of complex scenes with their unique typical elements. For example, in basketball courts **(a)**, our approach locates the basket (1st row), the backboard (2nd row), the jersey numbers (3rd row), the shot clock (4th row), a shoot (5th row), and the ball (6th row). Our approach can still focus and summarize the most critical elements even in more cluttered scenes like an outdoor stage, chemistry labs, or beer gardens. For example, in the case of outdoor stages **(f)**, we can see a lot of technical elements involved in their installation, including lights and top rails (1st row), monitor speakers (2nd row), microphones (4th row), and side rails (5th row).

each of the 10 selected countries to all others, resulting in 100K images, which we refer to as our parallel dataset. Performing translation using our finetuned model is critical for keeping scene elements consistent, as seen in Fig. 3c.

We show in supplementary material how performing semantic segmentation for each image and its translations to different countries enables measuring statistical trends. For example, we can measure that translations to Thailand or Brazil add many potted plants, and translations to Nigeria add dirt roads and people. We can visually confirm those trends on our parallel dataset.

Mining typical transformations across location. To further analyze our parallel dataset, we define a cross-location typicality measure to mine parallel translation of patches across locations. We define the co-typicality $\bar{\mathbf{T}}$ as the median typicality across location:

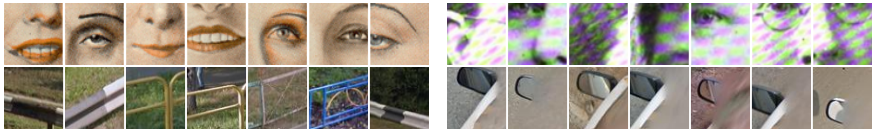
$$\bar{\mathbf{T}}(x) = \underset{c \in C}{\text{med}} \big[\mathbf{T}(x^{c_0 \to c}, c) \big], \tag{4}$$



(a) United States **(b) Russia** **(c) Brazil**

(d) France **(e) Japan** **(f) Thailand**

Fig. 8: Doersch *et al.*, 2013 [12] results on G^3 [28]. See text for details.



(a) Mixed Clusters **(b)** Mining Dataset Artifacts

Fig. 9: Limitations. The two most common failure modes we observe are: **(a)** issues in clustering, for example, clusters that contain diverse visual content, or multiple clusters that correspond to the same concept; **(b)** typicality highlighting artifacts of the dataset. Discovering artifacts is an expected behavior and can be useful for some applications.

where c_0 is the true label of the patch x and the median is computed over all countries in our set of 10 analyzed countries, C.

We can now ask: What visual elements are typical of a certain place and whose translation remains typical of another location? Instead of ranking single patches, we now rank a whole sequence of $|C|$ patches translated across locations according to $\bar{\mathbf{T}}$. We represent this sequence by concatenating the DIFT features of each patch [44]. To facilitate clustering, we first project the DIFT features of each patch from 1280 dimensions to 32 using UMAP [31]. To keep the same proportion of typical patches with respect to the number of analyzed images/sequences as in Section 4.3, we cluster the 10,000 visual elements with the highest co-typicality.

We display our results in Fig. 10, where for 6 selected clusters, we show in rows the four translated sequences closest to the cluster mean, highlighting in red the original image in each sequence. On the left column of Fig. 10, we show changes



Fig. 10: Clustering typical translations of elements across countries. Ranking translated visual elements according to $\bar{\mathbf{T}}$ and clustering the translated sequences results in groups of elements with similar variations. We show elements from 6 selected clusters out of 32. The source image for each sequence is highlighted in red. See text for details.

in typical architectural elements, such as gables, roofs, and windows. In contrast, on the right we show regulation-related elements, such as road tracks, utility poles, and license plates. Our approach allows us to both locate and visualize how common visual elements would vary from place to place, even though an exact match does not exist in the original data. For example, roofs typically turn dark brown when translated to the UK and black when translated to Japan.

5.2 Analysis of Medical Images

In Section 4.2, we discussed how typicality helps find relevant patches for an input label. In this section, we test this idea on completely different images: X-rays of patients who may suffer from a combination of various thorax diseases. We finetune Stable Diffusion on the ChestX-ray8 dataset [46] containing 108,948 frontal-view X-ray images annotated with 14 single-word disease-name labels. Experts annotated a test set of 879 images with 7 diseases with rectangular regions of interest (ROI) for each disease. For each image, we compute typicality per latent feature, interpolate the resulting typicality to the input dimension, and blur the resulting typicality map for visualization. In Fig. 11, we show the resulting typicality maps together with the ROI annotation before and after finetuning. Finetuning clearly improves the localization. We quantify this effect by computing the area under the precision recall-curve [5] (AUC-PR) associated

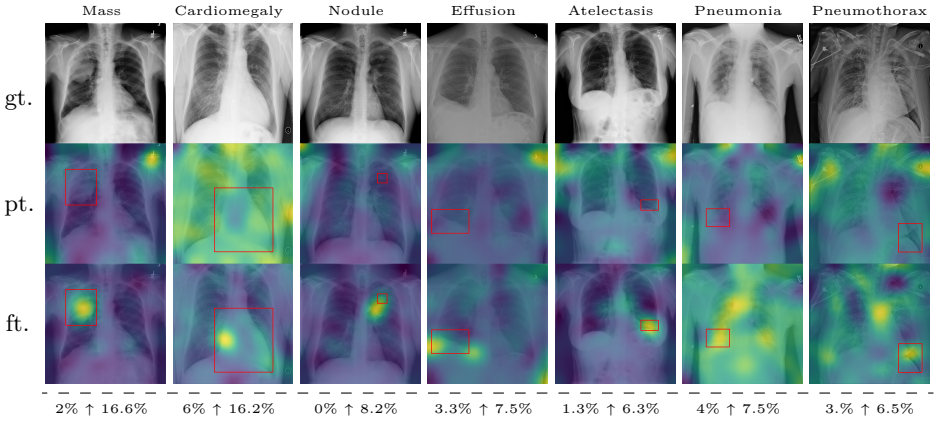


Fig. 11: Localizing abnormal areas in medical images. We visualize typicality when finetuning our model on the CXR8 dataset of thorax diseases [46]. After fine-tuning (ft.), we can see a clear focus of the typicality score on expert annotated areas (red boxes) for each disease, while initial predictions from the pretrained Stable Diffusion V1.5 model (pt.) are mostly noise. Images are ordered by AUC-PR after finetuning [5]. With ↑ we delimitate performance before and after finetuning, in the last row.

with the ROIs. As reported in Fig. 11, we see consistent improvement of this measure when finetuning the network (from 3.2% to 9.6%), ranging from +3.5% for Pneumonothorax (from 3% to 6%) to +14.6% for Mass (from 2% to 16.6%), which are respectively the least and most localized diseases. Similar to our other experiments, finetuning uses only image labels without localization supervision.

6 Conclusion

We presented a novel use of diffusion models as visual mining tools. We defined a typicality measure using a pretrained stable diffusion model finetuned for conditional image synthesis. We used typicality to mine visual summaries of four datasets, tagged by year or location. We further showed that we can use our typicality measure to localize abnormalities in medical data and extend it to discover trends in the variations of translated visual elements within a generated parallel dataset. In summary, our work presents a novel approach to visual data mining, enabling scaling to datasets significantly more extensive and diverse than those showcased in prior works, as demonstrated by our experiments.

Acknowledgments. This work was partially supported by the European Research Council (ERC project DISCOVER, number 101076028) and leveraged the HPC resources of IDRIS under the allocation AD011012905R1, AD0110129052 made by GENCI. It was also partially supported by ONR MURI. We thank Grace Luo for data, code, and discussion; Loic Landreu and David Picard for insights on geographical representations and diffusion; Carl Doersch, for project advice and implementation insights; Sophia Koepke for feedback on our manuscript.

References

1. geodummy. https://geodummy.com/, accessed: 2023-11-14
2. geoguessr. https://www.geoguessr.com/, accessed: 2023-11-14
3. geohints. https://geohints.com/, accessed: 2023-11-14
4. plonkit. https://www.plonkit.net/guide, accessed: 2023-11-14
5. Arun, N., Gaw, N., Singh, P., Chang, K., Aggarwal, M., Chen, B., Hoebel, K., Gupta, S., Patel, J., Gidwani, M., et al.: Assessing the trustworthiness of saliency maps for localizing abnormalities in medical imaging. Radiology: Artificial Intelligence (2021)
6. Azizi, S., Kornblith, S., Saharia, C., Norouzi, M., Fleet, D.J.: Synthetic data from diffusion models improves imagenet classification. TMLR (2023)
7. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. CVPR (2023)
8. Chai, L., Zhu, J.Y., Shechtman, E., Isola, P., Zhang, R.: Ensembling with deep generative views. CVPR (2021)
9. Chen, E.M., Sun, J., Khandelwal, A., Lischinski, D., Snavely, N., Averbuch-Elor, H.: What's in a decade? transforming faces through time. Computer Graphics Forum (2023)
10. Dalens, T., Aubry, M., Sivic, J.: Bilinear image translation for temporal analysis of photo collections. Transactions on Pattern Analysis and Machine Intelligence (2019)
11. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. NeurIPS (2021)
12. Doersch, C., Singh, S., Gupta, A., Sivic, J., Efros, A.A.: What makes paris look like paris? Communications of the ACM (2015)
13. Ginosar, S., Rakelly, K., Sachs, S.M., Yin, B., Lee, C., Krahenbuhl, P., Efros, A.A.: A century of portraits: A visual historical record of american high school yearbooks. IEEE Transactions on Computational Imaging (2017)
14. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626 (2022)
15. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NeurIPS (2020)
16. Ho, J., Saharia, C., Chan, W., Fleet, D.J., Norouzi, M., Salimans, T.: Cascaded diffusion models for high fidelity image generation. JMLR (2022)
17. Jae Lee, Y., Efros, A.A., Hebert, M.: Style-aware mid-level representation for discovering visual connections in space and time. ICCV (2013)
18. Jahanian, A., Puig, X., Tian, Y., Isola, P.: Generative models as a data source for multiview representation learning. ICLR (2022)
19. Kaoua, R., Shen, X., Durr, A., Lazaris, S., Picard, D., Aubry, M.: Image collation: Matching illustrations in manuscripts. ICDAR (2021)
20. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. Advances in Neural Information Processing Systems **35**, 26565–26577 (2022)
21. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. ICLR (2014)
22. Kotchemidova, C.: Why we say "cheese": Producing the smile in snapshot photography. Critical Studies in Media Communication (2005)
23. Lee, S., Maisonneuve, N., Crandall, D., Efros, A., Sivic, J.: Linking past to present: Discovering style in two centuries of architecture. IEEE International Conference on Computational Photography (ICCP) (2015)

24. Lee, Y.J., Efros, A.A., Hebert, M.: Style-aware mid-level representation for discovering visual connections in space and time. ICCV (2013)
25. Li, A.C., Prabhudesai, M., Duggal, S., Brown, E., Pathak, D.: Your diffusion model is secretly a zero-shot classifier. ICCV (2023)
26. Li, Y., Liu, H., Wu, Q., Mu, F., Yang, J., Gao, J., Li, C., Lee, Y.J.: GLIGEN: open-set grounded text-to-image generation. CVPR (2023)
27. Lloyd, S.: Least squares quantization in pcm. IEEE transactions on information theory (1982)
28. Luo, G., Biamby, G., Darrell, T., Fried, D., Rohrbach, A.: G^3: Geolocation via guidebook grounding. Findings of EMNLP (2022)
29. Luo, G., Dunlap, L., Park, D.H., Holynski, A., Darrell, T.: Diffusion hyperfeatures: Searching through time and space for semantic correspondence. NeurIPS (2023)
30. Matzen, K., Bala, K., Snavely, N.: StreetStyle: Exploring world-wide clothing styles from millions of photos. arXiv preprint arXiv:1706.01869 (2017)
31. McInnes, L., Healy, J., Saul, N., Grossberger, L.: Umap: Uniform manifold approximation and projection. The Journal of Open Source Software (2018)
32. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: Sdedit: Guided image synthesis and editing with stochastic differential equations. ICLR (2022)
33. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text inversion for editing real images using guided diffusion models. CVPR (2023)
34. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. ICML (2021)
35. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 (2022)
36. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. CVPR (2022)
37. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. CVPR (2023)
38. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. NeurIPS (2022)
39. Shen, X., Efros, A.A., Aubry, M.: Discovering visual patterns in art collections with spatially-consistent feature learning. CVPR (2019)
40. Shen, X., Efros, A.A., Joulin, A., Aubry, M.: Learning co-segmentation by segment swapping for retrieval and discovery. CVPR Image Matching workshop and Transformer workshop, 2022 (2021)
41. Shen, X., Pastrolin, I., Bounou, O., Gidaris, S., Smith, M., Poncet, O., Aubry, M.: Large-scale historical watermark recognition: dataset and a new consistency-based approach. ICPR (2021)
42. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. ICML (2015)
43. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. ICLR (2021)
44. Tang, L., Jia, M., Wang, Q., Phoo, C.P., Hariharan, B.: Emergent correspondence from image diffusion. NeurIPS (2023)
45. Tumanyan, N., Geyer, M., Bagon, S., Dekel, T.: Plug-and-play diffusion features for text-driven image-to-image translation. CVPR (2023)

46. Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., Summers, R.M.: Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. CVPR (2017)
47. Xu, J., Liu, S., Vahdat, A., Byeon, W., Wang, X., De Mello, S.: Open-vocabulary panoptic segmentation with text-to-image diffusion models. CVPR (2023)
48. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. ICCV (2023)
49. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)