CAPTAIN: Comprehensive Composition Assistance for Photo Taking

FARSHID FARHAT, MOHAMMAD MAHDI KAMANI, and JAMES Z. WANG*

Many people are interested in taking astonishing photos and sharing them with others. Emerging high-tech hardware and software facilitate the ubiquitousness and functionality of digital photography. Because composition matters in photography, researchers have leveraged some common composition techniques, such as the rule of thirds and the perspective-related techniques, in providing photo-taking assistance. However, composition techniques developed by professionals are far more diverse than well-documented techniques can cover. We present a new approach to leverage the underexplored photography ideas, which are virtually unlimited, diverse, and correlated. We propose a comprehensive fork-join framework, named CAPTAIN (Composition Assistance for Photo Taking), to guide a photographer with a variety of photography ideas. The framework consists of a few components: integrated object detection, photo genre classification, artistic pose clustering, and personalized aesthetics-aware image retrieval. CAPTAIN is backed by a large managed dataset crawled from a Website with ideas from photography enthusiasts and professionals. The work proposes steps to decompose a given amateurish shot into composition ingredients and compose them to bring the photographer a list of useful and related ideas. The work addresses personal preferences for composition by presenting a user-specified preference list of photography ideas. We have conducted many experiments on the newly proposed components and reported findings. A user study demonstrates that the work is useful to those taking photos.

CCS Concepts: • **Computer Vision** → **Computational photography**; • **Applied computing** → *Arts and humanities*; • **Computing methodologies** → *Object detection.*

Additional Key Words and Phrases: image aesthetics, deep learning, image retrieval, recommender system

1 INTRODUCTION

Digital photography is of great interest to many people, regardless of whether they are professionals or amateurs. It has been estimated that over a billion photos are taken every year and they are primarily taken with smartphones. People on social networks often share their photos with their friends. Smartphones' increasing computing power and ability to connect to more powerful computing platforms via the network make them potentially useful as a composition assistant to amateur photographers. Major smartphone manufacturers have started to introduce on-device photo enhancement capabilities.

Emerging technologies, including artificial intelligence (AI)-chips and AI-aware mobile applications, provide more opportunities for composition assistance. Taking stunning photos often needs expertise and experience at a level that professional photographers have. Like in other visual arts, a lack of a common alphabet similar to music notes or mathematical equations makes transferring knowledge in photography difficult. To many amateurs, as a result, photography is

*F. Farhat is with the School of Electrical Engineering and Computer Science, The Pennsylvania State University, USA (e-mail: fuf111@psu.edu). M. M. Kamani and J. Z. Wang are with the College of Information Sciences and Technology, The Pennsylvania State University, USA (e-mail: mqk5591@psu.edu; jwang@ist.psu.edu. This work used the Extreme Science and Engineering Discovery Environment (XSEDE) supported by National Science Foundation grant number ACI-1548562.

1551-6857/2021/1-ART1 $15.00
https://doi.org/10.1145/3462762

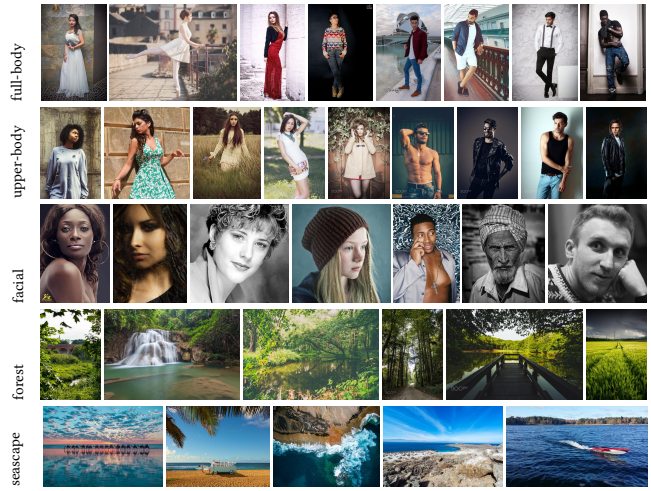


Fig. 1: Sample photos retrieved from the dataset based on the photo category and/or subject gender. Each retrieved result shows a collection of photography ideas that can be used by an amateur to compose photos for a given situation.

mysterious, and gaining skills is neither easy nor quick. Nonetheless, many people are fascinated about professional-quality photos and desire to have the ability to create similar-quality photos themselves for the scenes or events they are interested in. Because aesthetics in photography is strongly linked to human creativity, it is daunting for an AI to compose photographs at a given scene or a given studio setup that can impress people in a way professional photographers do. *We attempt to connect human creativity as demonstrated through their creative works with AI.*

Aesthetics and composition in photography have been heuristically explored as a collection of rules or principles such as balance, geometry, symmetry, the rule of thirds, and framing [29, 67]. It is well known that professional photographers take a large number of pictures, and through their practice, they gain experience and knowledge which in turn enable them to be creative [28]. Some composition rules or principles have been well articulated and many amateurs make use of them in their photo taking. However, the set of known principles can hardly cover the creativity and experience of thousands of photographers around the world. There is no unique photography idea for a given situation, and people have different opinions on those ideas depending on their cultural background, gender, age, experience, and emotional state. As a result, if the aesthetic quality of photos is quantified by one number, it can only emulate the average opinion of the general public, which may or may not be useful for a particular person.

It would be helpful if an AI can help people explore places [52] and select *photography ideas* from thousands of professional-quality photos for a given scene. The key technical difficulties for accomplishing this goal are (1) finding a suitable mapping between an amateurish photo of a scene and underlying professional-quality photography ideas, (2) handling a virtually unlimited number of photography ideas of a scene, and (3) providing meaningful and intuitive in-situ assistance to the photographer based on personal preference. Using a data-driven approach through recommendations from a large professional-quality dataset, our work tackles these challenges.

The multimedia and computer vision communities have been leveraging some of the photography composition principles for aesthetics assessment [14, 25, 38, 41, 70]. Other approaches manipulate the photo to comply with artistic rules, and they are referred to as auto-composition or re-composition. The techniques include smart cropping [45, 59, 72], warping [8, 35], patch re-arrangement [2, 12, 46], cutting and pasting [4, 75], and seam carving [18, 33]. However, they do not help an amateur photographer capture a more impressive photo to begin with. The arrangement

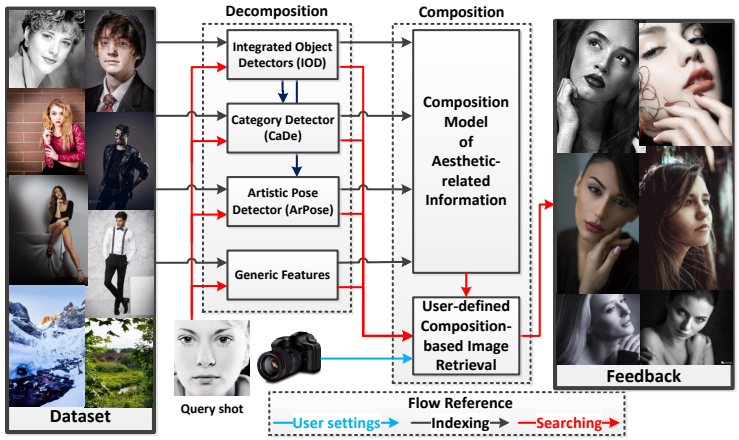


Fig. 2: The flowchart of our composition assistance framework: Blue, black and red flows show the user settings, indexing, and searching flows respectively. The decomposition box extracts the aesthetics-aware features and computes the composition model. The composition box retrieves well-composed images from our dataset for user-specified preferences.

of dark and light masses, which is known as "Notan" in visual art, has been used for composition analysis [32]. More recently, perspective-related techniques [78], the triangle technique [20], and portrait composition technique [17] have also been exploited.

We investigate a holistic framework for helping people take a better shot with regard to their current photography location and need. The framework addresses the differences in preferences of the users by adjusting the ranking process used to retrieve recommendation photos. After getting the first shot from the camera, our framework provides highly-scored related photos as pre-composed "recipes" (i.e. photography ideas) for the user to consider. As an example, regarding personalized criteria (such as photo category and subject gender), Figure 1 shows sample results retrieved from the photo dataset. These photos illustrate various locations, scenes, and categories. One can argue that while photos in the same row have the same category or gender, each photo has a photography idea(s) that is different from those used in other photos of the same row. For example, in the 2nd photo from the left in the 1st row, the subject using *rule-of-thirds* emphasizes the region of interest, and in the 2nd and 3rd photos from the right in the same row, the subjects cross their legs and bend one of the knees to form a *triangle* in the resulting photo. As mentioned before, the rule-of-thirds and triangle techniques are popular techniques used by professionals. Also, they may use one technique, but the way they use them is different, forming different photography ideas.

To address the complexity of transferring photography idea(s) to a user, we break down the scene that the user wants to take a photo from into composition primitives and then build them up for a better composed shot using professional-quality photos from the dataset with similar composition primitives. To accommodate the user's individual preferences, we perform personalized aesthetics-aware image retrieval (PAIR). Figure 2 shows the flowchart of our approach for assisting photographers in taking an improved photo. Based on the first query shot, highly-rated photos are retrieved from the collected dataset using the user-specified preferences (USP) and our composition model (CM). The **main contributions** of our work are as follows:

- We propose a new *fork-join framework* that understands the mapping between a photo and its potential underlying photography idea(s) through decomposing the taken photo into aesthetics/composition-related ingredients (Section 4) and followed by composing those ingredients to show recommended ideas to the photographer (Section 5). The framework leverages virtually unlimited number of photography ideas from professional-quality photographs.

- We design the *decomposition* step to extract composition primitives of a query shot using various detectors including the newly developed integrated object detector (IOD) (Section 4.1), category detector (CaDe) (Section 4.2), and our proposed artistic pose detector (ArPose) (Section 4.3). The IOD consists of a collection of performance-enhanced detectors including an object detector, a pose estimator, and a scene parser. The integration of them substantially boosts the detection accuracy by the proposed hysteresis fusion (Section 4.1.2). The CaDe has top-down decisive hierarchical clustering (Section 4.2.1) and multi-class categorization to leverage genre information (Section 4.2.2). The ArPose performs pose clustering (Section 4.3.1) to extract pose information using joint to line distance and skeleton context features.
- We address the complexity of transferring photography knowledge, caused by existence of abundant, diverse, and correlated photography ideas of a scene, by providing personalized meaningful and useful feedback to photographers. We design the *composition* step to get a similarity score (Section 5.1) and perform personalized aesthetics-aware retrieval (Section 5.2).
- In our framework, we manage a *dataset* containing 500K+ photos where 200K+ of them are highly rated covering a large number of photography ideas (Section 3) for training and retrieval. Using this dataset, we accommodate users' needs for composition, by showing a ranked list of photos based on user-specified preferences (USP).

2 RELATED WORK

While our approach to maximize photography idea coverage is novel, the work is closely related to the existing literature in aesthetic quality assessment, re-composition techniques, and composition rule-based feedback systems.

Aesthetic Quality Assessment: Basic image aesthetics and composition rules in visual art [28, 29, 67], including geometry, color palette, and the rule of thirds, have first been studied computationally by Datta *et al.* [14] and Ke *et al.* [25] as visual aesthetic features. Luo *et al.* [38], and Wong and Low [70] attempted to leverage a saliency map method, and considered the features of the salient parts because more appealing parts of an image often reside in the prominent region. Marchesotti *et al.* [41] showed that generic image descriptors were useful to assess image aesthetics, and built a generic dataset for composition assessment - the Aesthetic Visual Analysis (AVA) dataset [43]. Deep learning-based approaches [27, 36, 37, 40, 66] exploit customized architectures to train image aesthetic-quality models with annotated datasets, and the outcome is an estimation for actual (average) or personalized [55] aesthetic rating of an image.

Image Re-Composition: Auto-composition systems [3, 4] actively manipulate and then re-compose the taken photo for a better view. Cropping techniques [60, 62, 63, 69] separate the region of interest (ROI) with the help of a saliency map, an eye fixation, basic aesthetic rules [75], or visual aesthetics features in the salient region [45, 59, 72]. Warping [35] is another type of re-composition that represents an image as a triangular or quad mesh, to map the image into another mesh while keeping the semantics and perspective unchanged. Also, R2P [8] detects the foreground part in the reference image. Then, it re-targets the salient part of the image to the best-fitted position using a graph-based algorithm. Furthermore, patch re-arrangement techniques mend two ROIs in an image together. Pure patch rearrangements [2, 12, 46] detect a group of pixels on the border of the patch and match this group to the other vertical or horizontal group of pixels near the patched area. Also, cut-and-paste methods [4, 75] remove the salient part and re-paint the foreground with respect to the salient part and the borders, and then paste it to the desired position in the image. Another auto-composition system, seam carving [18, 33], replaces useless seams.

On-site Feedback Systems: An aesthetic assessor may find a metric to evaluate the aesthetic quality of an image, but the way it conveys this assessment to take a better photo is also crucial. In [9]

the authors search for views within a scene and retrieve a better crop relying on known structural features and visual saliency of professional photographs. There exist prior works that use the meta-data information (such as geo-location, weather, and time), and recommend a better position in the frame for standing people [44, 68], camera state guidance [50, 74], or quality and uniqueness-aware view cells [51] where their dataset contains known landmark photos. Combining the spring-electric graph model with color energy from visual arts, Rawat *et al.* [53] use visual balance as an image aesthetic factor to recommend social group position in a scene. An on-site aesthetics feedback system [32] retrieves similar images with known composition rules as qualitative feedback. But giving such feedback may be unrelated or unrealistic to the user, and the retrieved results may not be aesthetically useful to a photographer. More recently a perspective-related technique [78] and a triangle technique [20] retrieve similar photos to a query photo having perspective or triangle. Compared to the previous work in [17], we use a much broader dataset in terms of the size (about twice), diversity (portrait and landscape), quality (higher rating photos) of the dataset (Sections 3 and 6.1). Further, we implemented novel object detectors in Sections 4.1 and 6.2, novel category detection (Sections 4.2 and 6.2.5), novel artistic pose clustering (Sections 4.3 and 6.2.6). We also employed a novel integration of detectors (hysteresis fusion) in Sections 4.1.2 and 6.2.4, new decomposition and composition models with a faster retrieval system for user preferences and ranking (Sections 5 and 6.3). Also, our new method is extensively compared with recent deep learning-based methods for object detection [6, 11, 54, 56, 64, 76] and image retrieval [19, 61].

3 THE DATASET

The most valuable resource used by our framework is the collected dataset because it contains a large number of innovative photography ideas from around the world. We have examined photo-sharing websites for photography purposes including Flickr, Photo.net, DPChallenge, Instagram, Pinterest, and Unsplash, but none of them properly cover several categories such as full-body and upper-body in portrait photography as well as landscape photography ideas.

Portrait and Landscape Dataset: The dataset is gradually collected by crawling the 500px website which contains photos from millions of photographers around the world who are expanding their social networks of colleagues while exploiting technical and aesthetic skills to make money by marketing their photographs. To get the file list and then the images sorted by rating, we have implemented a distributed multi-IP address, block-free Python script. Nearly half a million images for the current dataset have been collected. The dataset has diverse photography ideas especially for the aforementioned portrait categories (full body, upper body, facial, group, couple or any two-body, side-view, hand-only, and leg-only) and landscape categories (nature, urban, etc) from highly-rated images taken by mostly photography enthusiasts and professionals. While Figure 1 shows sample photos from the dataset, Figure 6 in Section 6.1 shows the properties of the dataset. As a result, more than 90% of the images were viewed more than 100 times, and nearly half of the images in the dataset had a very high rating between 40 and 50, out of 60.

Automating Dataset Annotation: Our dataset contains 500K+ images where 200K+ of them are highly rated. We have manually annotated around 50K+ of the dataset for training, verification, and testing purposes. More precisely, we annotate 10K+ images for object detector, 5K+ for pose estimator, about 5K for scene parser, and about 25K for portrait. Also, we discard the rest of the training/testing set as they are unrelated. Then, we leverage multiple highly accurate detectors to automate and accelerate the annotation process of the rest of the images. However, the accuracy of our IOD (92.02%) and our CaDe (91.60%) for auto-annotation are high enough to retrieve aesthetics-aware exemplars. Also, the redundancy across our designed detectors makes the annotation process more robust.

4 PHOTO DECOMPOSITION

Content-based image retrieval (CBIR) methods help us map unbounded correlated data (*e.g.* an image) to a bounded range (*e.g.* feature vector), and then find similar images based on similarity metrics. But there are many restrictions to exploit them directly for applied problems. As mentioned before, there is *no limit* to innovation in visual arts. Hence, it is very difficult if not impossible for available methods to map an image to a set of useful and related photography ideas which are abundant, diverse, and correlated. Also, as the number of ideas increases, mean average precision (MAP) falls abruptly at the rate of $O(\frac{1}{n})$, and manual idea labeling of a large dataset is costly in terms of computational time and available budget.

To recommend better-composed photos to a photographer, we decompose the query image from the camera shot into composition ingredients called aesthetics-aware information. This information includes high-level features (such as semantic classes, photography categories, human pose classes, subject gender, and photo rating) as well as low-level features (such as color, texture, etc). To accelerate the retrieval process from the dataset based on a query image, we perform the decomposition procedure on all images in the dataset as an offline process, called *indexing*, shown as black arrows in Figure 2. We construct the composition model (CM) after indexing the whole dataset. If new images join the dataset, we index them and update our CM. In the *searching* step shown as red arrows in Figure 2, we decompose the query image, and compare it with our CM. Then, we retrieve the highly-ranked photos from the dataset based on the decomposed values of the query and user-specified preferences (USP).

Through this section, we describe the proposed integrated object detector (IOD) to determine semantic classes in a query image more comprehensively and more accurately than a single object detector. Also, the proposed category detector (CaDe) specifies the photography genre and style. Furthermore, the proposed artistic pose clustering (ArPose) extracts human pose information specifically for portrait photography.

4.1 Integrated Object Detectors (IOD)

To tackle the problem of classifying a virtually unlimited number of photography ideas, we need composition ingredients of a query shot, and then map them to top-ranked photography ideas. One of these ingredients is semantics inside query shot. To detect these semantics more accurately, we adopted deep-learning architectures and improved the detection accuracy compared to the state-of-art detectors by training our customized architecture on an augmented dataset including common failure cases (CFC) from our dataset, plus other available datasets including MSCOCO [34] and ADE20K [77]. Figure 3 illustrates how state-of-the-art object detector (YOLOv3 [54]), human pose estimator (OpenPose [7]), and scene parser (PSPNet [76]) poorly perform on a CFC set of images in our dataset compared to qualitative results from our IOD. Because OpenPose misses at facial photos to detect human parts like the neck in close-up photos and it is not very accurate at "two" or "group" categories to associate parts overlapping. Non-person detection of YOLOv3 under 34% probability is sometimes not reliable, and PSPNet detection is partially not accurate enough at photos with many objects, as it partitions the photo into small segments and it never considers overlapped area. To improve the accuracy of the detectors, we have changed the deep-learning architecture in terms of reduction layers, transformation parameters such as maximum rotation, crop size, scale min, and max. Because we can speed up the process by changing the reduction layers, and higher rotation and bigger portraits are more important in photography.

We adopted our object detector network architecture inspired by the GoogLeNet model [65] with 24 convolutional layers followed by fully connected layers, but with a simpler reduction to convolution layers to be faster. The output of the network is the bounding boxes of the detected objects with their probabilities. In our model, we do not consider non-person objects whose detection

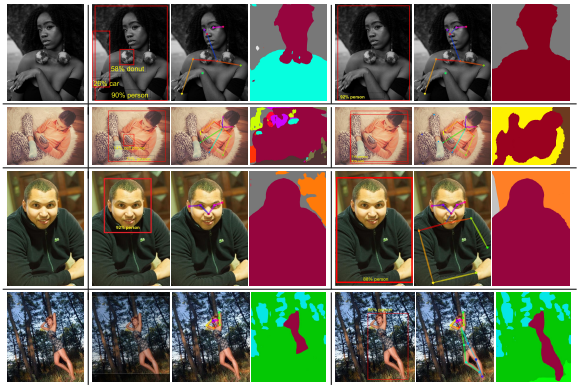


Fig. 3: Qualitative results show the improvement by our integrated object detector (IOD). Each row includes the original, YOLOv3, OpenPose, PSPNet, and the results from our IOD.

probability is less than 28%, because any wrong detection affects all pixels in the bounding box. As a result, we divide the input image into bigger chunks of 5×5 grid for higher accuracy, and smaller objects which are less important for detection as a secondary subject of photography. Our pose estimator architecture has a multi-stage convolutional neural network with two parallel lines predicting a limb confidence map and encoding the limb-to-limb association inspired by [7]. We adjust the transformation parameters of the architecture including maximum rotation degree to 60, crop size to 500, scale min to 0.6, and scale max to 1.0, since higher rotation degrees and bigger people are used frequently in our work. To design our scene parser, we ignore confusing labels like building and skyscraper. We place the related objects in the same object category. Also, our scene parser architecture exploits a 4-level pyramid pooling module [76] with sizes of 1×1, 2×2, 3×3 and 4×4 respectively. We do not consider detecting small objects in the scene since they are mostly not the main subject of the photographer.

4.1.1 Value Unification. To perform computation on the inference of our customized detectors in the next steps, we need to unify them in terms of pixel-level tensors. We define their scores as $-\log(1-p)$ for each pixel of the image. The object-ID and its score for each pixel are represented as an $m \times n \times 2$ tensor. Our object detector, scene parser, and pose estimator infer object-IDs respectively across 80 objects, 150 semantics, and 18 anatomical part IDs. Thus, for each image ($I_{m \times n}$) we have:

$$T^{I,\mathrm{od}}_{m \times n \times 2} = \left[t^{I,\mathrm{od}}_{i,j,k} \right], t^{I,\mathrm{od}}_{i,j,1} = C^{I,\mathrm{id}}_{i,j}, t^{I,\mathrm{od}}_{i,j,2} = -\log_2(1 - p^{I,\mathrm{od}}_{i,j}), \tag{1}$$

$$T^{I,\mathrm{sp}}_{m \times n \times 2} = \left[t^{I,\mathrm{sp}}_{i,j,k} \right], t^{I,\mathrm{sp}}_{i,j,1} = A^{I,\mathrm{id}}_{i,j}, t^{I,\mathrm{sp}}_{i,j,2} = -\log_2(1 - p^{I,\mathrm{sp}}_{i,j}), \tag{2}$$

$$T^{I,\mathrm{pe}}_{m \times n \times 2} = \left[t^{I,\mathrm{pe}}_{i,j,k} \right], t^{I,\mathrm{pe}}_{i,j,1} = J^{I,\mathrm{id}}_{i,j}, t^{I,\mathrm{pe}}_{i,j,2} = -\log_2(1 - p^{I,\mathrm{pe}}_{i,j}), \tag{3}$$

where I is an input image, m is the number of rows, n is the number of columns in the image, $T^{I,\mathrm{od}}$ is the corresponding tensor of object detector, $C^{I,\mathrm{id}}_{i,j} \in \{1..80\}$ is the object-ID of the pixel at (i,j), $p^{I,\mathrm{od}}_{i,j}$ is the object-ID probability of the pixel at (i,j), $T^{I,\mathrm{sp}}$ is the tensor of scene parser, $A^{I,\mathrm{id}}_{i,j} \in \{1..150\}$ is the object-ID of the pixel at (i,j), $p^{I,\mathrm{sp}}_{i,j}$ is the object-ID probability of the pixel at (i,j), $T^{I,\mathrm{pe}}$ is the tensor of pose estimator, $J^{I,\mathrm{id}}_{i,j} \in \{1..18\}$ is the joint-ID of the pixel at (i,j), and $p^{I,\mathrm{pe}}_{i,j}$ is the joint-ID probability of the pixel at (i,j).

4.1.2 Hysteresis Fusion. To expand the coverage of the photography idea space in the dataset, all detectable objects in each image using our detectors are leveraged. We maximize the dataset coverage while the accuracy is higher than 90%. Our *hysteresis fusion* optimizes the LOW and HIGH thresholds of the detection probability which is the average probability (or score in Eq. 1, 2, and 3) of the pixels of the object X for each (object X, detector Y) binary. If all detectors are below their LOW thresholds for object X, it means there is no object X in the image. If one of the detectors is above its HIGH threshold, it means there is an object X in the image. There is a narrow ambiguity region between LOW and HIGH values which covers a few images that we ignore.

To tune the thresholds, we conduct the experiments in Section 6.2.4 and consider detection probability as the object detector feature, and normalize area as the pose estimator feature. We get a bi-variate histogram (extendable to N-dimensional histogram for N detectors) in Figure 9 illustrating the frequency of the images smart-binned by the normalized object detector and pose estimator scores. Following these thresholds, the accuracy of our IOD scheme is 92.02% (higher than each detector), and 84.7% of the images are covered.

4.1.3 Object Importance. To prioritize the prominence of the objects in the image, we seek to use the importance map of the objects, because the subject of the image should be more important even if its detection probability is lower. To rank the order of the objects, we exploit the max score multiply by a saliency map (S) features with our centric distance (D) feature to get our weighted saliency map (W).

$$W^I(i,j) = \max\left(T^{I,\text{od}}_{*,*,2}, T^{I,\text{sp}}_{*,*,2}\right)_H S^I(i,j)D^I(i,j)\,, \tag{4}$$

$$D^I(i,j) = e^{-\|[i,j]-c^I\|_k}/K, \tag{5}$$

$$c^I = \frac{\sum_{i,j} S^I(i,j).[i,j]}{\sum_{i,j} S^I(i,j)}\,, \tag{6}$$

where $W^I(i,j)$ is our weighted saliency map point-wisely for image I, $max(.)_H$ operation is a hysteresis max on the second plane of the tensors (score matrix), $S^I(i,j)$ is a fast implementation of saliency map of image I [22], and $D^I(i,j)$ is our centric distance feature of image I, K is a tunable constant equal to $\sum_{i,j} e^{-\|[i,j]-c^I\|_k}$ for image I, the binary value c^I is the center of the mass coordinate, and $\|.\|_k$ is the k-th norm operator where $k = 1$ in our experiments. Our weighted saliency map makes the detected objects prioritized, because we sum up the scores from the semantic classes, and we end up with a total score for each semantic class. The output of this step is a weighted vector of detected semantic classes (undetected object has zero weight) in the query image. We show it as the following vector where the elements represent the importance (normalized as a probability) of the corresponding object in the image:

$$F_{I,\text{iod}} = \left[f^{\text{imp}}_1 \;\; f^{\text{imp}}_2 \;\; \dots \;\; f^{\text{imp}}_{210}\right]\,, \tag{7}$$

$$f^{\text{imp}}_k = \frac{\sum_{\forall \text{pixel(i,j) in obj(k)}} W(i,j)}{\sum_{\forall i,j} W(i,j)}\,, \tag{8}$$

where $\forall k \in \{1, ..., 210\}$, f^{imp}_k is the importance value of k-th object which is the summation of the weighted saliency of its every pixel (i,j), and $F_{I,\text{iod}}$ is the importance vector of all objects.

4.2 Category Detector (CaDe)

The photo categories in portrait include full-body, upper-body, facial, side-view, two (couple or two people), group (more than two people), faceless, headless, hand-only, and leg-only, which are ten classes. In landscape photography, there are sea, mountain, forest, cloud, and urban, which are five classes. While we focus on portrait and landscape photography genres, we believe that this work can be extended to other genres as well. Knowing the photo genres and categories helps our framework guide the photographer more adequately because it retrieves better-related results

based on the photographer preferences. The downside can be the low coverage or a limited number of contents on the leaves of this hierarchical tree of the photo styles, but our comprehensive dataset addresses this potential issue.

4.2.1 Top-down Hierarchical Clustering. To distinguish a portrait from a landscape photo, the number of people in the image is estimated by the max (union) number of person-IDs higher than their corresponding HIGH thresholds across the detectors in integrated object detector (IOD). If the score for detecting a person is lower than a LOW threshold for all detectors in IOD (intersection), there is no person in the image. Then, if there is a water-like, mountain-like, plant-like, cloud-like, or building-like object in the image with a total area higher than 26.5% (empirically tuned for landscape), the landscape category will be recognized as well. Otherwise, the image is ignored because its subject is not for portrait or landscape photo.

4.2.2 Portrait Multi-class Categorization. To automate an efficient and accurate portrait categorization, we formulate the problem as a multi-class error-correcting output code model using multiple support vector machine binary learners (let say ECOCSVM). The inputs are our feature vector and the corresponding class labels. Since we are using 10 portrait categories or unique class labels, it needs 55 ($= 10 \times (10 + 1)/2$) binary SVM learners with radial basis function (RBF or Gaussian) and a one-vs-one coding design. We have annotated 5% (about 25K+) of portrait photos uniformly selected at random from the dataset as the ground truth of the portrait categories. Then, we train an ECOCSVM with the feature vectors and the corresponding labels of 80% (about 20K) of our ground truth and leave the rest for testing our ECOCSVM. Our feature vector for each photo includes 40 different features as follows:

- General MAX: (1,2) max scores for detected people, and (3,4) max areas for the detected people from object detector and pose estimator.
- Intersection Area: (5) the area(s) of the people with the highest detection probability, (6,7) the scores of these people, (8,9) the areas of these people from object detector and pose estimator.
- Number of people: (10,11) number of people higher than the HIGH threshold for each detector, (12,13) number of people with area higher than 5% for each detector from object detector and pose estimator, (14) max of feature # 10 and feature # 11, (15) max of feature # 12 and feature # 13, (16) max of feature # 14 and feature # 15.
- Limb Features: (from 17 to 40) the limbs respectively including nose, neck, right shoulder, right elbow, right wrist, right hand, left shoulder, left elbow, left wrist, left hand, right hip, right knee, right ankle, right leg, left hip, left knee, left ankle, left leg, right eye, left eye, eyes, right ear, left ear, ears which add up to 40 features.

The output of this step for an image query is the following unitary vector that shows its category:

$$F_{I,\text{cade}} = [f_1^{\text{facial}} \; f_2^{\text{fullbody}} \; f_3^{\text{upperbody}} \; f_4^{\text{two}} \; f_5^{\text{group}} \; f_6^{\text{sideview}} \; f_7^{\text{leg}} \; f_8^{\text{noface}} \; f_9^{\text{hand}} \; f_{10}^{\text{nohead}}], \quad (9)$$

where $F_{I,\text{cade}}$ shows the unitary category vector of the image I by CaDe detector, and only one of the vector elements is one and the rest are zero. The mean average accuracy of our category detection is shown in Table 4 in Section 6.2.5 for the dataset images divided by various styles.

4.3 Artistic Pose Clustering (ArPose)

Posing, one of the essential ingredients of portrait photography, could substantially differentiate between amateur and professional shots. Having little experience in portrait photography, finding correct postures, or coming up with novel poses is hard for amateur photographers. Hence, it is vital for our system to have an understanding of different poses and how to categorize them.

Although RTMPPE extracts body joints in images, these joints are merely considered as our features for pose detection. We use two sets of features on top of joint coordinates to define the distance between different poses. These sets of features are scale-invariant, thus regardless of the

Fig. 4: Qualitative results of major clusters derived from our algorithm on the portrait dataset. Each row represents the top poses of each cluster.

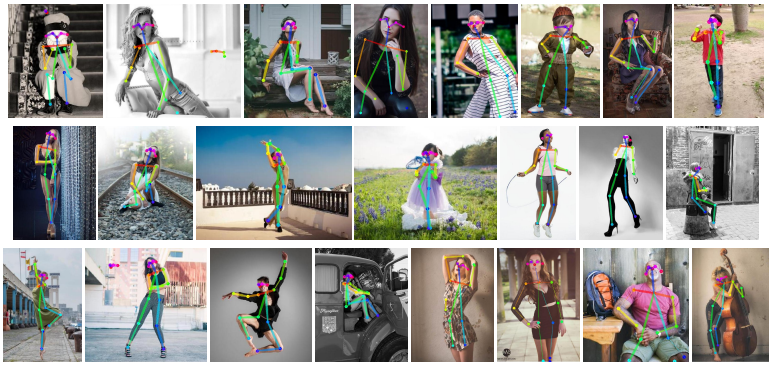


Fig. 5: Qualitative results of major clusters derived from the portrait dataset by DEC algorithm. Each row represents the top poses in each cluster. The DEC algorithm fails in clustering poses because the poses in the same cluster are not all consistent.

scale of the human body in images, we measure the similarity of two poses. These features are defined as follows:

- **Joint to Line Distance ($\mathbf{dist_{JtoL}}$)**: This distance vector consists of the distances between each joint and any line that connects two other joints. To have a scale-invariant distance, we normalize the distances with the maximum distance of each body in the image. Having the joint j_l and the line crossing two other joints, j_m and j_n, the $\text{dist}_{\text{JtoL}}$ is calculated as follows:

$$\text{dist}_{\text{JtoL}}(l, m, n) = 2S_{\Delta_{lmn}}/||j_m - j_n||_2, \tag{10}$$

  where $S_{\Delta_{lmn}}$ is the area under the triangle formed by three joints. Based on the total number of joints in each body, which is 18, and the total number of different distances is $18 \times \binom{17}{2} = 2448$.
- **Skeleton Context (SC)**: Also, another scale-invariant feature vector from previous work [23, 24] is leveraged. Skeleton context is a polar histogram of each point in the skeleton indicating the angular and distance distribution of other points in the skeleton around that point. We benefit from the angular distribution of each point and create an 18×18 angular matrix for each body in the image.

These features are designed to capture the relative position of each joint with respect to other points, hence, they are used as a measure of distance between different poses. We concatenate these features together to use the relative distance and polar information of both. Next, we use these features to cluster images based on various poses.

4.3.1 Pose Clustering. To rank the professional poses and find similar ones to the pose in the query image, we use clustering methods that distinguish body postures and group similar ones using the features explained in Section 4.3. To do so, we use clustering algorithms, k-means and Deep Embedding [71], and compare the results of these clustering methods. To determine the optimal number of clusters for the dataset, there are several heuristic methods including but not limited to *elbow* [26] and *silhouette* [57] methods. Having too many clusters would diminish the novelty and diversity of the results, in the sense that it tries to have samples as close as possible to one cluster. On the other hand, keeping the number of clusters low would affect the quality of clustering, such that irrelevant poses might appear in the same cluster. The result of our experiment using the elbow method shows that the optimal number of cluster heads is around 12-15 as depicted in Figure 10 in Section 6.2.6.

Then, we set up two clustering algorithms, k-means and Deep Embedding Clustering (DEC). For k-means, the only adjustable parameter is the number of clusters, but for DEC, we should set up the autoencoder network in addition to the number of clusters. As suggested by Xie *et al.* [71] and tested by ourselves, the network with 4 layers of encoder consisting of 500, 500, 2000, and 10 neurons in each unit performs astonishingly well on the clustering task of different supervised datasets including but not limited to MNIST [30], STL [13], and REUTERS [31]. Although DEC works great on these supervised datasets, it has not been tested on an actual unsupervised dataset, simply because there is not a gold standard to evaluate the performance on those datasets. However, visual data like the portrait dataset reveals how these algorithms perform, based on human eye evaluation of the results. Hence, we compare the results of this deep model for clustering with our feature-based k-means clustering. In k-means, to define the probability that each sample is in the cluster or the degree to which each sample belongs to a cluster, we use the same quantity in fuzzy C-means clustering [16]:

$$q_{ij}^{-1} = \sum_{k=1}^{K} \left(\frac{||x_i - c_j||_2}{||x_i - c_k||_2} \right)^{\frac{2}{m-1}}, \tag{11}$$

where x_i is the sample, c_j is the center of the cluster j, m is a positive real number greater than 1 which defines the smoothness of the function, and q_{ij} represents the probability that the sample belongs to the cluster. Also, DEC has defined a similar quantity [71] using Student's t-distribution:

$$q_{ij} = \frac{(1 + ||z_i - c_j||_2^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j'}(1 + ||z_i - c_{j'}||_2^2/\alpha)^{-\frac{\alpha+1}{2}}}, \tag{12}$$

in which z_i is the embedded version of x_i, and α is the degree of freedom in Student's t-distribution. Using these metrics we estimate the probability that each sample belongs to a cluster. Hence the output feature of the ArPose module would be in the form of:

$$F_{I,\text{arpose}} = \begin{bmatrix} f_{I,1}^{\text{arpose}} & f_{I,2}^{\text{arpose}} & \dots & f_{I,K}^{\text{arpose}} \end{bmatrix}^T, \tag{13}$$

where $f_{I,j}^{\text{arpose}} = q_{I,j}$ is the probability that the pose detected in image I belongs to the j^{th} cluster from our pool of K clusters as defined above.

The qualitative results of the k-means-based clustering algorithm in Figure 4 show the top-ranked poses in major clusters. Those of the DEC algorithm are in Figure 5. The images are ranked based on their probability computed in Eq. 11 and 12. As shown in the figures, k-means clusters surprisingly better than DEC, that is, different clusters distinguish different pose styles and each cluster represents visually almost similar pose. However, DEC fails to accomplish the goal of the clustering task based on human poses. Since the input features are intelligently chosen to be related to the goal, the input space is linearly separable, however, the result of the DEC shows information loss in the autoencoder. We tried the k-means algorithm with PCA to reduce the dimension of

the input space to 10 (as it is in the output of the autoencoder in DEC), and still the results of the k-means surpasses DEC's. Through that, we successfully cluster the portrait image and retrieve almost similar poses or novel ideas in that pose cluster based on the probability of the poses.

The other properties such as rating, tags, and gender in the shot are also extracted from the image and its descriptor. For the low-level features, we collect all 4096 generic descriptors via a public pre-trained CNN model [10] on ImageNet [15] and the conventional features of Mitro's method [42] as shown in the following equation. Note that there is no limit to collect any other aesthetics-aware information from query image to extend our work depending on photo genre.

$$F_{I,\text{vgg}} = \left[f_{I,1}^{\text{vgg}} \;\; f_{I,2}^{\text{vgg}} \;\; \cdots \;\; f_{I,4096}^{\text{vgg}} \right]^T, \tag{14}$$

where $F_{I,\text{vgg}}$ is a vector containing generic features of image I, and $f_{I,i}^{\text{vgg}}$ $\forall i$ is i-th generic feature. The superscript "T" represents the transpose of the vector/matrix. Also, we extract available statistical data via the image properties including rating, view counts, and gender. Then, we have them as follows:

$$F_{I,\text{stat}} = \left[f_{I,1}^{\text{rating}} \;\; f_{I,2}^{\text{views}} \right], \tag{15}$$

$$F_{I,\text{gender}} = \left[f_{I,1}^{\text{male}} \;\; f_{I,2}^{\text{female}} \;\; f_{I,3}^{\text{unknown}} \right], \tag{16}$$

where $F_{I,\text{stat}}$ is a vector containing the statistical data of image I including its rating $f_{I,1}^{\text{rating}}$ and its view counts $f_{I,2}^{\text{views}}$. Furthermore, $F_{I,\text{gender}}$ is a vector containing the gender specification of image I represented by [1 0 0] as male, [0 1 0] as female, or [0 0 1] as unknown.

4.4 Construction of Composition Model

To aesthetically index all photos in our dataset and easily search by composition features, we decompose them into the following feature vectors and construct our composition model (CM). In fact, $F_{I_i,\text{vgg}}$, $F_{I_i,\text{iod}}$, $F_{I_i,\text{cade}}$, $F_{I_i,\text{arpose}}$ and other aesthetics-aware information for all ($\forall i$) images are extracted and appended to corresponding matrices respectively including deep-learned generic features M_{vgg} (for color, texture, and edges), all detected objects M_{iod}, photography category M_{cade}, artistic pose M_{ap}, statistical features M_{stat} and detected gender M_{gnd}.

$$F_{I_i} = \left[F_{I_i,\text{vgg}} \; F_{I_i,\text{iod}} \; F_{I_i,\text{cade}} \; F_{I_i,\text{ap}} \; F_{I_i,\text{stat}} \; F_{I_i,\text{gnd}} \right], \tag{17}$$

where $i \in \{1, ..., N\}$, F_{I_i} is the feature vector of the image I_i. Then, we compute the corresponding feature matrix.

$$M_{\text{feat}}^T = \left[F_{I_1,\text{feat}}^T \; F_{I_2,\text{feat}}^T \cdots F_{I_N,\text{feat}}^T \right], \tag{18}$$

$$M = \left[M_{\text{vgg}} \; M_{\text{iod}} \; M_{\text{cade}} \; M_{\text{ap}} \; M_{\text{stat}} \; M_{\text{gnd}} \right], \tag{19}$$

$$M^T = \left[F_{I_1}^T \; F_{I_2}^T \; \ldots \; F_{I_N}^T \right], \tag{20}$$

where $(.)^T$ is transpose operation, "feat" is feature type from the set {vgg, iod, cade, ap, stat, gnd}, matrix M_{feat} is the corresponding feature matrix containing feature vector of each image in each row. The final feature matrix M is the composition model matrix which is the concatenation of all feature matrices or equivalently all feature vectors.

5 COMPOSITION OF VISUAL ELEMENTS

Basically, image retrieval methods want to optimize [5, 21] or customize [39] the process of retrieving images with similar semantics in specified regions, which is not an image aesthetics nor composition-related procedure. For example, an amateurish image may focus on a less important semantic as a photo subject or assign a region of interest to a less important semantic, but the retrieval results from our professional-quality dataset are free of such mistakes.

The goal of our composition step is to gather all composition elements from the previous step and recommend related photography ideas from our collected dataset satisfying personal preferences and aesthetics-aware information of the query image. The input to this step is the decomposed values of the image query and user-specified preferences (USP) with our composition model (CM). The output is a collection of well-composed images from the dataset recommended to the user. For example, if we focus on portraits, we desire feedback that contains well-posed portraits with similar semantics and category but better composition.

As we have collected a dataset containing generally well-composed images, we should dig into the dataset and look for images with "pretty" similar color, pattern, category, pose, or object constellation where the term "pretty" is framed by USP to address the user's needs and subjectivity. The existence of this professional-quality dataset makes it possible that the retrieved photos have highly accepted photography ideas by the people. Our image retrieval system is not supposed to find images with exactly similar colors, patterns, or poses, but it finds images with a better composition having similar semantic classes, category, or pose. Thus, the location of the movable objects does not matter, but the detected objects are important.

5.1 Similarity Scores and Normalization

Having our composition model for all images in our dataset and the query image, we first calculate the similarity score between the query image and any image in the dataset across the detectors. The similarity metric of generic VGG features (14) is the multiplication of the matrix M_{vgg} by the query vector F_{vgg}. Similarly, the category detector has a matrix by vector multiplication. For integrated object detectors, we use the Gaussian function after masking unrelated objects. For statistics and gender information, it is formulated as follows:

$$S_{\text{vgg}}(I,Q) = F_{I,\text{vgg}}^T F_{Q,\text{vgg}} \,, \tag{21}$$

$$S_{\text{cade}}(I,Q) = F_{I,\text{cade}}^T F_{Q,\text{cade}} \,, \tag{22}$$

$$S_{\text{iod}}(I,Q) = e^{-(\sum(F_{I,\text{iod}} \circ \text{sign}(F_{Q,\text{iod}}) - F_{I,\text{iod}}))^2} \,, \tag{23}$$

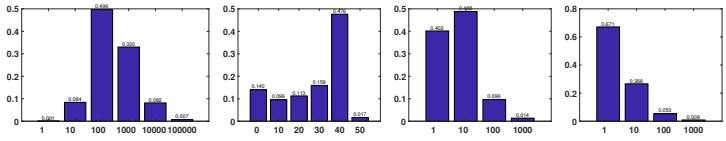
$$S_{\text{stat}}(I,Q) = |f_{I,1}^{\text{rating}} - f_{Q,1}^{\text{rating}}| \,, \tag{24}$$

$$S_{\text{gender}}(I,Q) = \begin{cases} 1, & \text{if } F_{I,\text{gender}} = F_{Q,\text{gender}} \\ -1, & \text{otherwise} \end{cases} \tag{25}$$

where F^T means the transpose of F, e is a mathematical constant about 2.72, the $\circ$ operation is the element-wise multiplication, sign(.) is the sign function operating on each element separately. Also, $S_{\text{vgg}}(I,Q)$, $S_{\text{cade}}(I,Q)$, $S_{\text{iod}}(I,Q)$, and $S_{\text{stat}}(I,Q)$ are similarity score values between image I and image Q respectively for generic CNN descriptors, category detection, integrated object detectors, and statistics and gender information. The similarity score function is easily generalized to a function between two different sets of images, *i.e.*, $I_{m\times 1}$ and $Q_{n\times 1}$ can be a set of images not only one image, and the output will be an $m \times n$ matrix. Since we want to score the similarity between the images in our dataset (say $\mathbb{I}$) and an image query (Q), in the above equations, vector $F_{I,\text{det}}^t$ will be substituted by matrix M_{det}, and the output will be a similarity vector, while "det" can be any detector $\in$ {vgg, iod, cade, arpose, stat, gender}.

To make the scores uniform across various detectors, we normalize each detector score vector by dividing by the summation of the whole output. Thus, each detector's similarity score is like a probability distribution over all images. We have:

$$S_{\text{feat}}^N(\mathbb{I},Q) = \frac{S_{\text{feat}}(\mathbb{I},Q)}{\sum_{i\in\mathbb{I},q\in Q} S_{\text{feat}}(i,q)} \,, \tag{26}$$



(a) view count. (b) rating. (c) vote count. (d) favorite count.

Fig. 6: The dataset properties: (a) the logarithmic distribution of the view counts, (b) the distribution of the ratings, (c) the logarithmic distribution of the vote counts, and (d) the logarithmic distribution of the favorite counts.

where $S^N_{\text{feat}}(\mathbb{I}, Q)$ is a normalized similarity score matrix between each image in $\mathbb{I}$ and each image in Q for detector feat $\in$ {vgg, iod, cade, arpose, stat, gender}. Also, we combine the similarity scores across various detectors to create a tensor of similarity scores for each pair of images from $(\mathbb{I}, Q)$. We have:

$$S^N(\mathbb{I}, Q) = \left[S^N_{\text{vgg}} \; S^N_{\text{iod}} \; S^N_{\text{cade}} \; S^N_{\text{arpose}} \; S^N_{\text{stat}} \; S^N_{\text{gender}} \right] , \tag{27}$$

where $S^N(\mathbb{I}, Q)$ is a tensor of size $d \times m \times n$ where d is the number of detectors (||feat|| here is 6), m is the number of images in $\mathbb{I}$, and n is the number of images in Q.

5.2 User Preferences and Ranking

Prior works have explored feature-based, example-based, and list-based personalized ranking systems for amateur photographs using conventional aesthetic qualities and personal preferences [73]. The example-based and list-based approaches are non-scalable when we have a large dataset as is in our approach. To rank the exemplar list, we multiply user-specified preferences as a probability vector containing the importance weights with our compositional primitive feature matrix. Adjusting weight vector can also cover decision tree-based ranking approaches as well. Now, we have:

$$W_{\text{USP}} = \left[W_{\text{vgg}} \; W_{\text{iod}} \; W_{\text{cade}} \; W_{\text{arpose}} \; W_{\text{stat}} \; W_{\text{gender}} \right]^T , \tag{28}$$

where W_{USP} is a $d \times 1$ vector showing the weights of the user for each detector, and "T" shows the transpose operation. Then, to retrieve the highest-ranked candidates as the results, the normalized similarity score matrix is multiplied by the USP vector. Consequently, we have:

$$V_{\text{pref}}(\mathbb{I}, Q) = W^T_{\text{USP}} S^N(\mathbb{I}, Q) , \tag{29}$$

where $V_{\text{pref}}(\mathbb{X}, Q)$ is the user's preferred image vector, and if we find the K top-ranked entries with respect to the vector values, the indexes of these entries represent the best high-quality recommendations to the image query (Q). The results for some queries with USP are shown in Figure 11 separated by the query image, the baseline retrieval with equal weights, and the retrieval with proper USP.

6 EXPERIMENTS

In the following sub-sections, we describe our experimental results which are categorized into different components of our method including (i) the dataset, (ii) the decomposition step, and (iii) the composition step. Furthermore, the decomposition step has multiple parts to demonstrate the effectiveness of our method compared to an available state-of-the-art or our baselines.

6.1 Dataset Properties

We have collected the images in the portrait and landscape categories from 500px Website and saved them as smaller images where their highest dimension has been resized to 500 pixels. Then, we have collected available metadata for each image including the number of views, the average ratings, the number of vote clicks, and the number of favorite clicks. We conduct statistical experiments to

Method	MAP	person	seat	plant	animal	car
YOLOv3[54]	51.5	72.5	39.4	32.8	69.7	54.0
Faster R-CNN[56]	52.7	73.9	40.6	34.2	71.0	54.2
Ours	60.1	77.8	53.1	46.6	69.5	57.7

Table 1: The accuracy comparison between our object detector model versus the YOLOv3 and Faster R-CNN on our dataset to detect some known objects.

Method	MAP	hea	sho	elb	wri	hip	kne	ank
OpenPose[6]	70.2	86.5	79.8	71.2	62.6	68.3	63.9	60.4
HRNet[64]	73.7	92.0	84.4	73.5	66.3	71.3	65.6	61.0
Ours	75.6	92.5	86.8	75.8	66.1	71.7	68.6	64.6

Table 2: The comparison between our pose estimator versus OpenPose and HRNet on our dataset to detect body parts.

get the properties of the collected dataset. Because some of these properties change dramatically in linear scale, their trends are captured intuitively better in the logarithmic x-axis. Figure 6 illustrates the distributions of the view counts, ratings, vote counts, and favorite counts of the dataset. Each bar represents a bin where its interval is from the corresponding number written under the bin to right before the number written under the next bin. Figure 6 shows that most of the images have been seen more than 100 times, i.e., 500px Website has a live community while many images have at least 1-10 votes or favorite clicks. Having a rating higher than 10 is considered high because the rating trend changes its slope direction from bin 0-9 to bin 10-19 negatively, and after that, the slope will positively grow until bin 40-49. Most of the images in the dataset have a rating of more than 40 which is a very high rating, and it indicates that the 500px community of the photographer has many highly-rated photos.

6.2 Decomposition Analysis

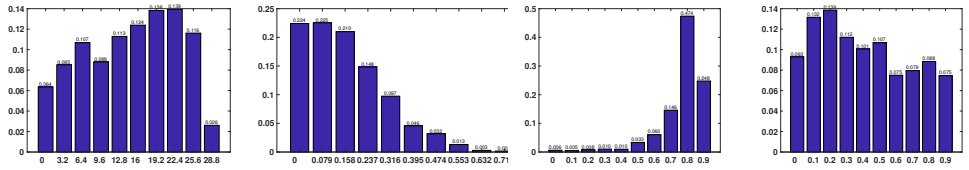
To show the effectiveness of our decomposition step, we conduct the following experiments on the object detector, the human pose estimator, and the scene parser used in our framework. Also, we examine the hysteresis fusion, the category detector, and the pose clustering.

6.2.1 Object Detection. Our object detector network contains 24 convolutional layers with two fully connected layers (mentioned in Section 4.1). We train it on ImageNet [15] and 8K+ images from our dataset, and three times on an annotated subset of 768 common failure cases (CFC) from our dataset. We evaluate and compare our model with YOLOv3 [54] and Faster R-CNN [56] on a test set of 2K+ images from our dataset. We use the regular MAP on all intended objects. Table 1 shows the MAP and the average accuracy of some objects (person, seat, plant, animal, and car) for our trained model versus YOLOv3 model. The "seat" average accuracy is the average for "seat, bench, and chair", "plant" average accuracy is the average for "plant, tree, and grass", and "animal" average accuracy is the average for "bird, cat, dog, cow, and sheep".

6.2.2 Pose Estimator. We train our pose estimator model on MSCOCO[34], MPII[1], 4K+ images from our dataset, and three times on our 317 CFC. To evaluate the performance of our pose estimator model on a test set of 1K+ images from our dataset, we leverage MAP of all limbs. The comparison results of the MAP performance between OpenPose[6], HRNet[64], and our approach on a subset of 507 testing images from our dataset are shown in Table 2, where the left limb and the right limb are merged.

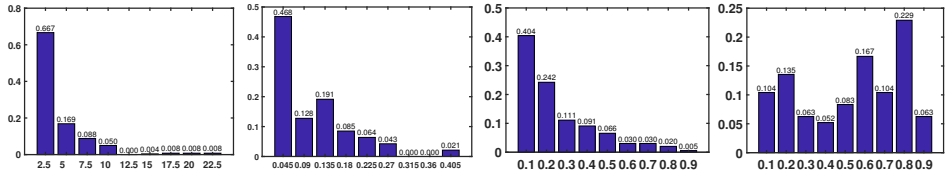
Method	Pixel Accuracy (%)	Mean IoU (%)
PSPNet[76]	74.9	40.8
DeepLab[11]	77.6	42.2
Ours	79.2	43.8

Table 3: The accuracy comparison between our scene parser versus PSPNet and DeepLab with 101-depth ResNet on our dataset.



(a) pose score (b) pose area (c) object probability (d) object area

Fig. 7: The distributions of (a) the score obtained from the pose estimator, (b) the normalized area obtained from the pose estimator, and (c) the detection probability obtained from the object detector, and (d) the normalized area obtained from the object detector for our ground-truth images with a "person" as a common object by the pose estimator and the object detector.



(a) pose score (b) pose area (c) object probability (d) object area

Fig. 8: For the ground-truth images without any "person", the distributions of (a) the highest score (if any) obtained from the pose estimator, (b) the highest normalized area of the highest score object (if any) obtained from the pose estimator, and (c) the detection probability for the dominant object (if any) obtained from the object detector, and (d) the normalized area of the dominant object (if any) obtained from the object detector.

6.2.3 Scene Parser. To train our scene parser model, we use the ADE20K dataset [77], 4K+ images from our dataset, and 576 common failure cases annotated by LabelMe [58]. To evaluate scene parsing performance on a test set of 1K+ images from our dataset, pixel-wise accuracy (PixAcc) and mean of class-wise intersection over union (CIoU) are measured. The performance values of our scene parser model versus PSPNet [76] and DeepLab[11] with ResNet-101 are shown in Table 3 which indicates better PixAcc and CIoU is achieved on our dataset.

6.2.4 Hysteresis Fusion. The coverage of the photography ideas is improved by hysteresis fusion which allows the union of all images above HIGH thresholds across the detectors. We show how we configure these tunable thresholds, while we trade-off between the coverage and the accuracy across the object detectors. When we have more than one detector with common detectable objects, we fuse multiple features from the detectors to enhance common object detection. For example, "person" is a common object between object detector and pose estimation. We perform our pose estimator on our ground-truth images with a person or without any person from the dataset, and calculate (a) the detection score (as mentioned in Eq. 3) and (b) the normalized area (*i.e.* the detected object area divided by the image area) of the dominant person (*i.e.* the person with the highest score) detected in each image as our pose estimator features. Also, we perform our object detector

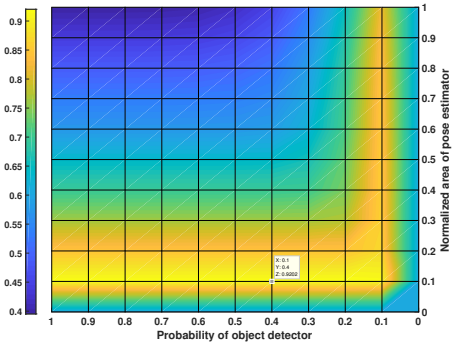


Fig. 9: The 2D MAP surface w.r.t the normalized area of the pose estimator and the detection probability of the object detector as a heat map.

on those images and compute (c) the detection probability and (d) the normalized areas of the dominant person (*i.e.* the person with the highest probability) detected in each image as our object detector features.

The distributions of the features obtained from our pose estimator and our object detector for "person" as a common object for the pose estimator and the object detector has been shown in Figure 7. In some images, no person is detected by the pose estimator and the object detector, because the pose estimator or the object detector has a detection error or there is no person in the image. We consider such detection as non-person object detection. Figure 8 shows the distributions of those features obtained from our pose estimator and our object detector when there is no person in our ground-truth images, but they detect a person. We have removed the frequency of the first component, *i.e.*, score or area = 0, from all of the curves in Figure 8, because the probability of zero score/area is very high and we want to bold the probabilities of the other score/area values.

Figure 7a shows pose estimator's score does not have enough sensitivity to detect a person, because the distribution is similar to a uniform probability mass function (PMF). Similarly, Figure 7d shows object detector's normalized area does not have enough sensitivity to detect a person, because the distribution is pretty uniform. But, the object detector's probability in Figure 7c and the pose estimator's normalized area in Figure 8b are not similar to a uniform distribution, and we can infer the cut-off thresholds from them. First, we derive the 2D probability density function (PDF) of these mutual features including the normalized area by the pose estimator and the detection probability by the object detector. Second, we determine the 2D MAP surface w.r.t these two parameters as a heat map. Finally, we search on the heat map to find the optimal point for these two mutual features. As shown in Figure 9, it can be inferred from the 3D histogram of these two features that the optimal HIGH cut-off thresholds are object detector's probability 40% and pose estimator's normalized area 10%. Similarly, the LOW thresholds are object detector's probability 28% and pose estimator's normalized area 4.5% that leads to the dataset coverage 84.7% and the detection accuracy 92.02% which is higher than any other detector accuracy solely.

6.2.5 Portrait Category Detection. As mentioned in Section 4.2, we start with top-down hierarchical clustering to specify the genre of the input image, and then we do multi-class categorization for portrait images. We train our model having 40 suggested features on a set of 20K+ annotated portraits from our dataset, and we test the model on another set of 5K+ annotated portraits. The mean average accuracy of the model is listed in Table 4 categorized by various styles.

Also, we just consider the first 16 features for object detectors including general max and number of detected people in the image as mentioned in Section 4.2 and train a model using the same ground truth as before. The current model is our baseline model because it can be used for any

	facial	full-body	group	hand	leg	noface	side view	two	upper-body
Our CaDe	96.13	94.52	89.80	61.73	80.42	84.64	75.02	78.57	92.93
16-feat Baseline	66.58	80.24	68.35	N/A	N/A	N/A	N/A	59.42	55.28

Table 4: The accuracy results of our category detector (CaDe) for ground-truth images compared to a 16-feature-based baseline.

Extractor	Effect	Baseline	% over baseline
IOD	object-aware	max(detectors)	17.30
CaDe	category-aware	16-feat version	32.54
ArPose	pose-aware	DEC	25.7

Table 5: The summary of the methods used in CAPTAIN with respect to extraction algorithm, effect on retrieval, and improvement over baseline.

Fig. 10: The result of the elbow method on the dataset. We could spot the elbow around 13-17 clusters. We are also showing the first derivative of the distortion, to show where it is going to flatten out.

other object detector, as the features can be defined in other object detector domains as well. To compare rationally with this baseline, we test the same set of images from our ground truth. The second line in Table 4 listed the baseline results. Because we remove limb features, the baseline cannot detect sub-genres such as hand-only, leg-only, no-face, and side view.

6.2.6 Artistic Pose Clustering. Regarding artistic pose clustering, we conduct an experiment to cluster similar professional poses using our features explained in Section 4.3. We do the clustering with a various number of cluster heads, and we find the optimal number of cluster heads for our dataset using the elbow method [26]. That being said, we use the elbow method and do the clustering 40 times with the different number of clusters ranged from 1 to 40. This method calculates the sum of squared errors (the distance of each point to the center of its cluster) and it is expected to see an elbow pattern in the plot of this error when the number of clusters is increasing. The result of this method on our dataset is depicted in Figure 10, which indicates that the best choice for the number of clusters in this dataset is between 13 and 17. Since we integrate many features with different extraction algorithms, Table 5 summarize these features in aspects like extraction method, effects, the improvement over baseline.

6.3 User Study for Composition

There exists no directly similar or comparable system in the literature to compare with our proposed framework. The studies [48–51, 53, 68] in related work have different goals because they should get a known landmark with its meta-data to extract photos with similar geo-location, weather, and time, and process them to find the best view or camera parameters. But we retrieve exemplars from our 500K+ portrait and landscape dataset based on aesthetics-aware primitives of the given photo including IOD semantics, VGG generic, CaDe, pose, and gender features.

To fairly evaluate the functionality and performance of our method, and measure how much the recommended photos are relevant to the query and helpful to the photographer, we conduct a

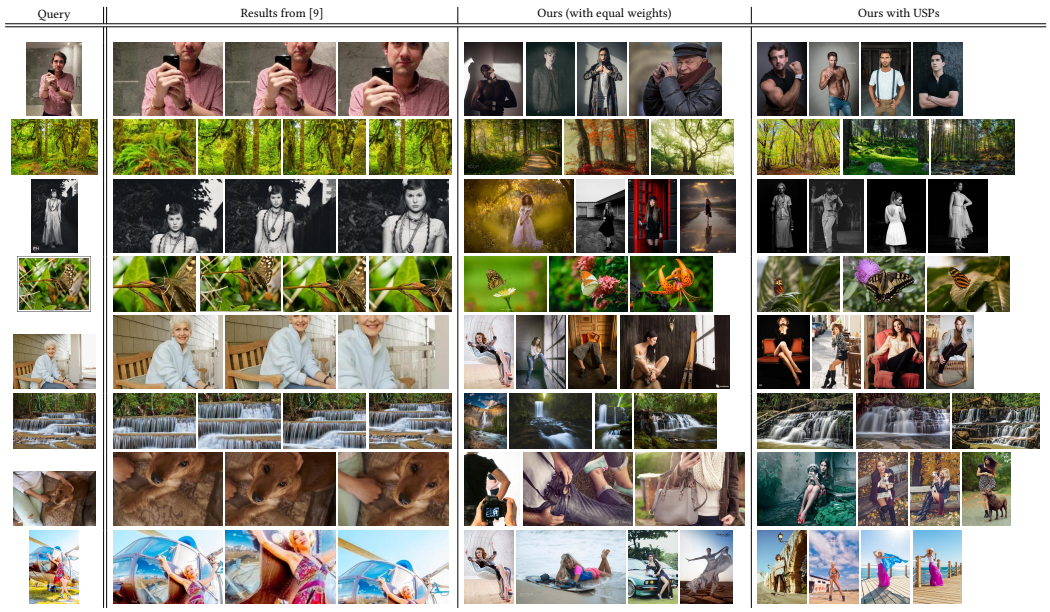


Fig. 11: Qualitative results of the composition step compared to [9]. Each row shows query image, results from [9], our method with equal weights, and ours with user-specified preferences (USP). The USP with respect to each row is: (1) $W_{\text{cade}} = W_{\text{gender}} = 0.5$, (2) $W_{\text{cade}} = W_{\text{vgg}} = W_{\text{iod}} = 0.33$, (3) $W_{\text{gender}} = W_{\text{vgg}} = W_{\text{cade}} = 0.33$, (4) $W_{\text{cade}} = W_{\text{iod}} = 0.5$, (5)$W_{\text{cade}} = W_{\text{gender}} = W_{\text{iod}} = 0.33$, (6) $W_{\text{cade}} = W_{\text{vgg}} = W_{\text{iod}} = 0.33$, (7) $W_{\text{cade}} = W_{\text{iod}} = 0.5$ and (8) $W_{\text{cade}} = W_{\text{vgg}} = W_{\text{iod}} = 0.33$.

quantitative user study to compare our method with other reasonable approaches. The first method directly finds good composition from scenes (called Chang's method [9]). The second method is a retrieval method based on the color, shape, and texture features (called Mitro's method [42]). The third and fourth ones are from a state-of-the-art semantic and scene retrieval method [47] with better convolutional networks (CNN) including the VGG-19 model [61] and ResNet-152 [19]. To create the last two baselines, all generic descriptors of the last pooling layer pre-trained on ImageNet [15] are evaluated for our dataset images as well as the features of the non-CNN-based method [42], and it is used as feature matrix M_{feat} in Eq. 20. The similarity scores and normalization are calculated following the composition step (section 5), and user preferences are specified uniform across all competitors. The qualitative results of the composition step for some queries with user-specified preferences (USP) are illustrated in Figure 11. Each row includes query image, results from method [9], our method for equal weights, and ours for USP-aware retrieval.

We select a variety of image queries (Figure 12) based on background scene and semantics, single versus group, full-body, upper-body, facial, standing versus sitting, and male versus female. We do not use USP-aware queries as shown in Figure 11, and we focus on the diversity of image queries. Also, the same question throughout the study is asked, and therefore, we do not convey any side information. Using a PHP-based website with usage guidance, the outputs of the methods are randomly shown in each row to be chosen by 103 participants.

The expected value of the accepted recommended photos by the participants with respect to the total number of recommendations including the baselines is 65.74%. More accurately, the histogram of the acceptability rate for the queries of the user study is shown in Figure 13. The x-axis shows the acceptability rate ranged from 0 to 1 with 0.1-width bins, *i.e.*, what percentage of the participants has accepted our recommended photos for those queries. The y-axis shows the

Fig. 12: A subset of image queries used for user study based on different types of categories such as semantics, single vs group, full-body, upper-body, facial, male vs female, etc.

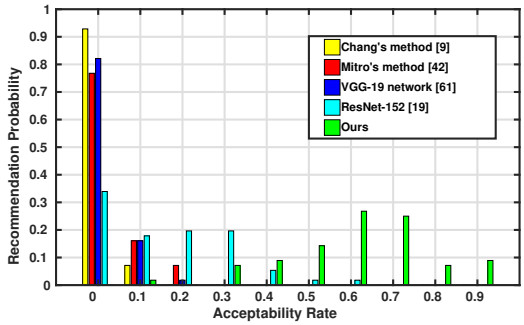


Fig. 13: The histogram of the acceptability rate based on the recommended photos versus the total number of recommendations (recommendation probability) compares ours with other methods.

frequency of our accepted recommendations by the total number of the examined corresponding queries (*i.e.* probabilities) which fall into each bin. The histogram has indicated that 16.07% of our recommended photos were accepted by over 80% of the participants, 67.86% of them with over 60%. Consequently, the majority of our recommended photos are accepted with a mean of 65.74%.

7 CONCLUSIONS

We have introduced a new framework for composition assistance that guides amateur photographers to capture better shots by providing exemplars. We have experimented with the proposed approach using a large dataset for portrait and landscape photography ideas that we have collected. This study leverages the integration of deep-learning-based detectors, hysteresis fusion, portrait categorization, and artistic pose clustering which makes the whole process automatic. As the number of photography ideas increases, retrieving the exemplars from the dataset becomes more challenging. Furthermore, the retrieval system not only finds similar images but also searches for images with similar semantic constellations with better composition through decomposition and composition steps. The performance of our framework has been evaluated by a set of experiments including comparisons with some competitors and a user study.

REFERENCES

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2014. 2D human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Columbus, OH, USA, 3686–3693.

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan Goldman. 2009. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics* 28, 3 (2009), 1–24.

[3] Subhabrata Bhattacharya, Rahul Sukthankar, and Mubarak Shah. 2010. A framework for photo-quality assessment and enhancement based on visual aesthetics. In *Proceedings of the ACM International Conference on Multimedia*. ACM, New York, NY, USA, 271–280.

[4] Subhabrata Bhattacharya, Rahul Sukthankar, and Mubarak Shah. 2011. A holistic approach to aesthetic enhancement of photographs. *ACM Transactions on Multimedia Computing, Communications, and Applications* 7, 1 (2011), 1–21.

[5] Yue Cao, Mingsheng Long, Jianmin Wang, and Shichen Liu. 2017. Deep visual-semantic quantization for efficient image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Honolulu, HI, USA, 1328–1337.

[6] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2019. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 43, 1 (2019), 172–186.

[7] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime multi-person 2D pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Honolulu, HI, USA, 7291–7299.

[8] Hui-Tang Chang, Po-Cheng Pan, Yu-Chiang Frank Wang, and Ming-Syan Chen. 2015. R2P: Recomposition and Retargeting of Photographic Images. In *Proceedings of the ACM International Conference on Multimedia*. ACM, Brisbane, Australia, 927–930.

[9] Yuan-Yang Chang and Hwann-Tzong Chen. 2009. Finding good composition in panoramic scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*. IEEE, Kyoto, Japan, 2225–2231.

[10] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Return of the Devil in the Details: Delving Deep into Convolutional Nets. In *Proceedings of the British Machine Vision Conference*. BMVA Press.

[11] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 40, 4 (2017), 834–848.

[12] Taeg Sang Cho, Moshe Butman, Shai Avidan, and William T Freeman. 2008. The patch transform and its applications to image editing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Anchorage, AK, USA, 1–8.

[13] Adam Coates, Andrew Ng, and Honglak Lee. 2011. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, Vol. 15. PMLR, Fort Lauderdale, FL, USA, 215–223.

[14] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. 2006. Studying aesthetics in photographic images using a computational approach. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 288–301.

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 248–255.

[16] J. C. Dunn. 1973. A Fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics* 3, 3 (1973), 32–57.

[17] Farshid Farhat, Mohammad Mahdi Kamani, Sahil Mishra, and James Z. Wang. 2017. Intelligent portrait composition assistance: integrating deep-learned models and photography idea retrieval. In *Proceedings of the ACM Conference on Multimedia, Thematic Workshops*. ACM, Mountain View, CA, USA, 17–25.

[18] YW Guo, M Liu, TT Gu, and WP Wang. 2012. Improving photo composition elegantly: Considering image similarity during composition optimization. *Computer Graphics Forum* 31, 7 (2012), 2193–2202.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 770–778.

[20] Siqiong He, Zihan Zhou, Farshid Farhat, and James Z Wang. 2018. Discovering Triangles in Portraits for Supporting Photographic Creation. *IEEE Transactions on Multimedia* 20, 2 (2018), 496–508.

[21] Ahmet Iscen, Yannis Avrithis, Giorgos Tolias, Teddy Furon, and Ondrej Chum. 2018. Fast spectral ranking for similarity search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Salt Lake City, UT, USA, 7632–7641.

[22] Laurent Itti, Christof Koch, and Ernst Niebur. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 20, 11 (1998), 1254–1259.

[23] Mohammad Mahdi Kamani, Farshid Farhat, Stephen Wistar, and James Z Wang. 2016. Shape matching using skeleton context for automated bow echo detection. In *Proceedings of the International Conference on Big Data*. IEEE, Washington, DC, USA, 901–908.

[24] Mohammad Mahdi Kamani, Farshid Farhat, Stephen Wistar, and James Z Wang. 2018. Skeleton matching with applications in severe weather detection. *Applied Soft Computing* 70 (2018), 1154–1166.

[25] Yan Ke, Xiaoou Tang, and Feng Jing. 2006. The design of high-level features for photo quality assessment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 1. IEEE, New York, NY, USA, 419–426.

[26] David J Ketchen and Christopher L Shook. 1996. The application of cluster analysis in strategic management research: an analysis and critique. *Strategic Management Journal* 17, 6 (1996), 441–458.

[27] Shu Kong, Xiaohui Shen, Zhe Lin, Radomir Mech, and Charless Fowlkes. 2016. Photo aesthetics ranking network with attributes and content adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, Cham, Germany, 662–679.

[28] Bert Krages. 2012. *The Art of Composition.* Skyhorse Publishing, New York, NY, USA.
[29] David Lauer and Stephen Pentak. 2011. *Design Basics.* Wadsworth Publishing, Belmont, CA, USA.
[30] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.
[31] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research* 5, Apr (2004), 361–397.
[32] Jia Li, Lei Yao, and James Z Wang. 2015. Photo composition feedback and enhancement. In *Mobile Cloud Visual Media Computing*. Springer, Cham, Germany, 113–144.
[33] Ke Li, Bo Yan, Jun Li, and Aditi Majumder. 2015. Seam carving based aesthetics enhancement for photos. *Signal Processing: Image Communication* 39 (2015), 509–516.
[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, Cham, Germany, 740–755.
[35] Ligang Liu, Yong Jin, and Qingbiao Wu. 2010. Realtime Aesthetic Image Retargeting. *Computational Aesthetics* 10 (2010), 1–8.
[36] Zhenguang Liu, Zepeng Wang, Yiyang Yao, Luming Zhang, and Ling Shao. 2018. Deep active learning with contaminated tags for image aesthetics assessment. *IEEE Transactions on Image Processing* (2018).
[37] Xin Lu, Zhe Lin, Hailin Jin, Jianchao Yang, and James Z Wang. 2015. Rating image aesthetics using deep learning. *IEEE Transactions on Multimedia* 17, 11 (2015), 2021–2034.
[38] Yiwen Luo and Xiaoou Tang. 2008. Photo and video quality evaluation: Focusing on the subject. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, Berlin, Heidelberg, 386–399.
[39] Long Mai, Hailin Jin, Zhe Lin, Chen Fang, Jonathan Brandt, and Feng Liu. 2017. Spatial-semantic image search by visual feature synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Honolulu, HI, USA, 1121–1130.
[40] Long Mai, Hailin Jin, and Feng Liu. 2016. Composition-preserving deep photo aesthetics assessment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Las Vegas, NV, USA, 497–506.
[41] Luca Marchesotti, Florent Perronnin, Diane Larlus, and Gabriela Csurka. 2011. Assessing the aesthetic quality of photographs using generic image descriptors. In *Proceedings of the International Conference on Computer Vision (ICCV)*. IEEE, Barcelona, Spain, 1784–1791.
[42] Joani Mitro. 2016. Content-based image retrieval tutorial. *arXiv preprint arXiv:1608.03811* (2016).
[43] Naila Murray, Luca Marchesotti, and Florent Perronnin. 2012. AVA: A large-scale database for aesthetic visual analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Providence, RI, USA, 2408–2415.
[44] Bingbing Ni, Mengdi Xu, Bin Cheng, Meng Wang, Shuicheng Yan, and Qi Tian. 2013. Learning to photograph: A compositional perspective. *IEEE Transactions on Multimedia* 15, 5 (2013), 1138–1151.
[45] Jaesik Park, Joon-Young Lee, Yu-Wing Tai, and In So Kweon. 2012. Modeling photo composition and its application to photo re-arrangement. In *Proceedings of the IEEE Conference on Image Processing*. IEEE, Orlando, FL, USA, 2741–2744.
[46] Yael Pritch, Eitam Kav-Venaki, and Shmuel Peleg. 2009. Shift-map image editing. In *Proceedings of the International Conference on Computer Vision (ICCV)*, Vol. 9. IEEE, Kyoto, Japan, 151–158.
[47] Filip Radenović, Giorgos Tolias, and Ondřej Chum. 2018. Fine-tuning CNN image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 41, 7 (2018), 1655–1668.
[48] Yogesh Singh Rawat. 2015. Real-time assistance in multimedia capture using social media. In *Proceedings of the ACM International Conference on Multimedia*. ACM, New York, NY, USA, 641–644.
[49] Yogesh Singh Rawat and Mohan S Kankanhalli. 2014. Context-based photography learning using crowdsourced images and social media. In *Proceedings of the ACM International Conference on Multimedia*. ACM, New York, NY, USA, 217–220.
[50] Yogesh Singh Rawat and Mohan S Kankanhalli. 2015. Context-aware photography learning for smart mobile devices. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 12, 1s (2015), 1–24.
[51] Yogesh Singh Rawat and Mohan S Kankanhalli. 2016. Clicksmart: A context-aware viewpoint recommendation system for mobile photography. *IEEE Transactions on Circuits and Systems for Video Technology* 27, 1 (2016), 149–158.
[52] Yogesh Singh Rawat, Mubarak Shah, and Mohan S Kankanhalli. 2019. Photography and Exploration of Tourist Locations Based on Optimal Foraging Theory. *IEEE Transactions on Circuits and Systems for Video Technology* 30, 7 (2019), 2276–2287.
[53] Yogesh Singh Rawat, Mingli Song, and Mohan S Kankanhalli. 2017. A spring-electric graph model for socialized group photography. *IEEE Transactions on Multimedia* 20, 3 (2017), 754–766.
[54] Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).

[55] Jian Ren, Xiaohui Shen, Zhe Lin, Radomír Mech, and David J Foran. 2017. Personalized Image Aesthetics. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, Venice, Italy, 638–647.
[56] S Ren, K He, R Girshick, and J Sun. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 39, 6 (2017), 1137.
[57] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20 (1987), 53–65.
[58] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. 2008. LabelMe: A database and web-based tool for image annotation. *International Journal of Computer Vision (IJCV)* 77, 1-3 (2008), 157–173.
[59] A. Samii, R. Měch, and Z. Lin. 2015. Data-driven automatic cropping using semantic composition search. *Computer Graphics Forum* 34, 1 (2015), 141–151.
[60] Anthony Santella, Maneesh Agrawala, Doug DeCarlo, David Salesin, and Michael Cohen. 2006. Gaze-based interaction for semi-automatic photo cropping. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. ACM, New York, NY, USA, 771–780.
[61] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. CNN features off-the-shelf: An astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, Columbus, OH, USA, 806–813.
[62] Fred Stentiford. 2007. Attention based auto image cropping. In *Proceedings of the International Conference on Computer Vision Systems*.
[63] Bongwon Suh, Haibin Ling, Benjamin B Bederson, and David W Jacobs. 2003. Automatic thumbnail cropping and its effectiveness. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*. ACM, New York, NY, USA, 95–104.
[64] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. 2019. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Long Beach, CA, USA, 5693–5703.
[65] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Boston, MA, USA, 1–9.
[66] Hossein Talebi and Peyman Milanfar. 2018. Nima: Neural image assessment. *IEEE Transactions on Image Processing* 27, 8 (2018), 3998–4011.
[67] Roberto Valenzuela. 2012. *Picture Perfect Practice: A Self-training Guide to Mastering the Challenges of Taking Photographs*. New Riders, Indianapolis, IN, USA.
[68] Patricia P Wang, Wei Zhang, Jianguo Li, and Yimin Zhang. 2008. Online photography assistance by exploring geo-referenced photos on MID/UMPC. In *Workshop on Multimedia Signal Processing*. IEEE, 6–10.
[69] Zijun Wei, Jianming Zhang, Xiaohui Shen, Zhe Lin, Radomír Mech, Minh Hoai, and Dimitris Samaras. 2018. Good view hunting: learning photo composition from dense view pairs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Salt Lake City, UT, USA, 5437–5446.
[70] Lai-Kuan Wong and Kok-Lim Low. 2009. Saliency-enhanced image aesthetics class prediction. In *Proceedings of the IEEE Conference on Image Processing*. IEEE, Cairo, Egypt, 997–1000.
[71] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, New York, NY, USA, 478–487.
[72] Jianzhou Yan, Stephen Lin, Sing Kang, and Xiaoou Tang. 2013. Learning the change for automatic image cropping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 971–978.
[73] Che-Hua Yeh, Brian A Barsky, and Ming Ouhyoung. 2014. Personalized photograph ranking and selection system considering positive and negative user feedback. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 10, 4 (2014), 1–20.
[74] Wenyuan Yin, Tao Mei, Chang Wen Chen, and Shipeng Li. 2013. Socialized mobile photography: Learning to photograph with social context via mobile devices. *IEEE Transactions on Multimedia* 16, 1 (2013), 184–200.
[75] Mingju Zhang, Lei Zhang, Yanfeng Sun, Lin Feng, and Wei-ying Ma. 2005. Auto cropping for digital photographs. In *Proceedings of the IEEE Conference on Multimedia and Expo*. IEEE, Amsterdam, Netherlands, 4–pp.
[76] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Honolulu, HI, USA, 2881–2890.
[77] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2017. Scene parsing through ADE20K dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Honolulu, HI, USA, 633–641.
[78] Zihan Zhou, Farshid Farhat, and James Z Wang. 2017. Detecting dominant vanishing points In natural scenes with application to composition-sensitive image retrieval. *IEEE Transactions on Multimedia* 19, 12 (2017), 2651–2665.