What Can Human Sketches Do for Object Detection?

Pinaki Nath Chowdhury  Ayan Kumar Bhunia  Aneeshan Sain  Subhadeep Koley
Tao Xiang  Yi-Zhe Song
SketchX, CVSSP, University of Surrey, United Kingdom.
{p.chowdhury, a.bhunia, a.sain, s.koley, t.xiang, y.song}@surrey.ac.uk

Abstract

Sketches are highly expressive, inherently capturing subjective and fine-grained visual cues. The exploration of such innate properties of human sketches has, however, been limited to that of image retrieval. In this paper, we for the first time, we cultivate the expressiveness of sketches but for the fundamental vision task of object detection. The end result is a sketch-enabled object detection framework that detects based on what you sketch – that “zebra” (e.g., one that is eating the grass) in a herd of zebras (instance-aware detection), and only the part (e.g., “head” of a “zebra”) that you desire (part-aware detection). We further dictate that our model works without (i) knowing which category to expect at testing (zero-shot) and (ii) not requiring additional bounding boxes (as per fully supervised) and class labels (as per weakly supervised). Instead of devising a model from the ground up, we show an intuitive synergy between foundation models (e.g., CLIP) and existing sketch models build for sketch-based image retrieval (SBIR), which can already elegantly solve the task – CLIP to provide model generalisation, and SBIR to bridge the (sketch→photo) gap. In particular, we first perform independent prompting on both sketch and photo branches of an SBIR model to build highly generalisable sketch and photo encoders on the back of the generalisation ability of CLIP. We then devise a training paradigm to adapt the learned encoders for object detection, such that the region embeddings of detected boxes are aligned with the sketch and photo embeddings from SBIR. Evaluating our framework on standard object detection datasets like PASCAL-VOC and MS-COCO outperforms both supervised (SOD) and weakly-supervised object detectors (WSOD) on zero-shot setups. Project Page: https://pinakinathc.github.io/sketch-detect

1. Introduction

Sketches have been used from prehistoric times for humans to express and record ideas [35, 76]. The level of expressiveness [28, 41] they carry remains unparalleled today even in the face of language [14, 82] – recall that moment that you want to resort to pen and paper (or Zoom Whiteboard) to sketch down an idea?

Sketch research has also flourished over the past decade [16, 70, 90, 99], with a whole spectrum of works on traditional tasks such as classification [30] and synthesis [15, 31, 54], and those more sketch-specific such as modelling visual abstraction [1, 59], style transfer [74] and continuous stroke fitting [17], to cute applications such as turning a sketch into a photo classifier [5, 37].

The expressiveness of sketches, however, has been only explored in the form of sketch-based image retrieval (SBIR) [19, 70, 94], especially the fine-grained [2, 6, 9] variant (FG-SBIR). Great strides have been made, with recent systems already reaching maturity for commercial adaptation [6] – a great testimony to how cultivating sketch expressiveness can make a real impact.

In this paper, we ask the question – what can human sketches do for the fundamental vision tasks of object detection? The envisaged outcome is, therefore, a sketch-enabled object detection framework that detects based on what you sketch, i.e., how you want to express yourself. Sketching a “zebra eating the grass” (in Fig. 1) should detect “that...
zebra from a herd of zebras (instance-aware detection), and it will also give you the freedom to be specific with parts (part-aware detection), so if the “head” of a “zebra” is what you would rather desire, then just sketch the very head.

Instead of devising a sketch-enabled object detection model from the ground up, we show that an intuitive synergy between foundation models (e.g., CLIP [64]) and off-the-shelf SBIR models [8,100] can already, rather elegantly, solve the problem – CLIP to provide model generalization, and SBIR to bridge the (sketch→photo) gap. In particular, we adapt CLIP to build sketch and photo encoders (branches in a common SBIR model) by learning independent prompt vectors [42,71] separately for both modalities. More specifically, during training, the learnable prompt vectors are prepended into the input sequence of the first transformer layer of CLIP’s ViT backbone [22] while keeping the rest frozen. As such, we inject model generalization into the learned sketch and photo distributions. Next, we devise a training paradigm to adapt the learned encoders for object detection, such that the region embeddings of detected boxes are aligned with the sketch and photo embeddings from SBIR. This allows our object detector to train without requiring additional training photos (bounding boxes and class labels) from auxiliary datasets.

To make our sketch-based detector more interesting (general-purpose [13,57]), we further dictate it also works in a zero-shot manner. For that, following [10], we extend object detection from a pre-defined fixed-set setup to an open-vocab setup. Specifically, we replace the classification heads in object detectors with prototype learning [49], where the encoded query sketch features act as the support set (or prototypes). Next, the model is trained under the weakly supervised object detection (WSOD) setting [10,78], using a multi-category cross-entropy loss over the prototypes of all possible categories or instances. However, while SBIR is trained using object-level (single object) sketch/photo pairs, object detection works on image-level (multiple categories). Hence, to train object detectors using SBIR, we also need to bridge the gap between object and image-level features. Towards this, we use a data augmentation trick that is embarrassingly simple yet highly effective for robustness towards corruption and generalisation to out-of-vocab [101,102] – we randomly select \( n = \{1, \ldots, 7\} \) photos from SBIR datasets [30,75] and arbitrarily tile them on a blank canvas (similar to CutMix [101]).

In summary, our contributions are (i) for the first time cultivating the expressiveness of human sketches for object detection, (ii) sketch-based object detector that detects what you intend to express in your sketch, (iii) an object detector that is both instance-aware and part-aware, in addition to performing conventional category-level detection, (iv) a novel prompt learning setup to marry CLIP and SBIR to build the sketch-aware detector that works without needing bounding box annotations (as supervised [67]), class labels (as weakly supervised [10]), and in a zero-shot manner. (v) results outperform both supervised (SOD) and weakly supervised object detectors (WSOD) on zero-shot setup.

2. Related Works

Sketch for Visual Understanding Hand-drawn sketches serve as a useful query modality for visual understanding tasks that involve human perception and structural cues. Sketches not only convey a visual description [35] but also exhibit artistic styles [103]. This makes sketch a vital querying modality for the creative industry, like artistic image editing [98] and animation [93]. Unlike photos that are passively captured by a camera, sketches are actively drawn by humans, which makes them a good visual representation [4,7] enriched with human participation. Apart from the widely explored sketch-based image retrieval [2,9,70,74], sketch as a query has shown potential in several vision understanding tasks like incremental learning [5] image and video synthesis [31,44,54], representation learning [1,88], image-inpainting [92], 3D shape retrieval [96], 3D shape modelling [15], medical image analysis [43,90], object localisation [69,86] and segmentation [38,63].

Studying sketch as a query for object detection by Tripathi et al. [86], several limitations surfaced with respect to problem definition as well as architectural designs. Firstly, instead of fine-grained matching, sketch was used to specify object category (easier via text/keyword [29,64]), thus overlooking the potential of sketch to model fine-grained details. Secondly, it requires both bounding-box and sketch annotation, which increases the annotation budget without significant improvement in performance over traditional object detection setups. Thirdly, due to an expensive annotation, only fewer than 50% object categories in existing object detection datasets [23,52] are available for training. Finally, using an early fusion strategy [95] of sketch with object detection results in recomputing object regions for each new sketch – leading to a slower detection framework with increasing query sketches. In this paper, we propose a fine-grained sketch-based object detection framework that uses only object-level sketch photo pairs without any bounding-box annotations for training and is scalable with multiple fine-grained query sketches, even under zero-shot setup.

Supervised Object Detection Object detection jointly localises and identifies objects in an image. Traditional object detectors rely on large datasets such as PASCAL VOC [23] and MS-COCO [52], containing thousands of examples per object category which are quite time-consuming to annotate, unlike our pipeline, that leverage sketch-photo pairs as annotation. Existing literature on object detection is bifurcated as: (i) fast yet less accurate single-shot [51,55,65,66,108]; (ii) slow but more accurate two-stage object detectors [26,27,32,67]. To fully exploit the fine-
grained cues provided by sketch, our proposed method aligns with the accurate two-stage detectors that predict object regions using selective search in RCNN [27], ROI pooling [26] in Fast-RCNN, and Region Proposal Network (RPN) with ROIAAlign in Faster-RCNN [67]. While there has been several attempts with sophisticated architectural modifications [46, 108, 109], Faster-RCNN [67] still acts a fundamental building block for several downstream tasks like scene-graph generation [97], visual grounding [58], and relationship prediction [110]. Therefore, we resort to the more traditional Faster-RCNN based two-stage pipeline.

**Weakly Supervised Object Detection (WSOD)** Collecting bounding box annotation per object category is already a time-consuming process, which is aggravated even further by fine-grained object detection (e.g., recognising animal species). To overcome this, existing WSOD adopt two schools of thought: (a) formulate this as a multiple instance learning (MIL) [10,20,21,40,48,79,85,104] problem that interpret an image as a bag of proposals or regions. The image is labelled positive if one of the regions contains the object of interest; otherwise negative. (b) CAM-based methods [105,106] that use class activation maps to predict proposals. Specifically, an image is fed to a backbone network to generate a feature vector from which the bounding box of each class is predicted by thresholding activation maps with the highest probability. Although CAM-based methods are faster, we use MIL-based technique as it can detect multiple instances within the same category [78].

**Data Augmentation in Computer Vision** Data augmentation improves the sufficiency and diversity of training data. Approaches vary from simple image rotation and flipping to more advanced techniques of image erasing [11] like CutOut [18], Hide-and-Seek [81] and image mixing like MixUp [102] and CutMix [101]. Aiming to generalise sketch-based object detection to complex scenes while training exclusively on existing object-level sketch photo pairs [75], we employ a CutMix [101] like data augmentation trick – a method that replaces removed sub-regions with a patch from another image to synthesise new images.

**Sketch-Based Object Representation** Sketch with its intrinsic ability to model fine-grained visual details makes it an ideal modality for object retrieval, giving rise to avenues like category-level [16,19,70,94,99] and fine-grained (FG) instance-level [2,6,9,72] sketch-based image retrieval (SBIR). Contemporary research also explored zero-shot SBIR [19, 73, 99], cross-domain translation [61] and approaches like reinforcement learning based on-the-fly retrieval [9], self-supervised [3, 62], etc. Apart from object-level images, retrieving sketched objects from scene images was studied using graph convolutional networks [53] and optimal transport [12]. Similar to cross-category FG-SBIR [8,61], here we explore fine-grained sketch photo association for object detection by adapting large vision-language models like CLIP [64] using prompt engineering [107].

### 3. Proposed Method

**Overview** We propose a new paradigm training object detection without bounding box annotation or image-level class labels. Instead, we use sketch-based image retrieval for supervision. This leads to several emergent behaviours (i) fine-grained object detection – specify focused region-of-interest using fine-grained visual cues in sketch. (ii) category-level object detection – specify the category of detected instances via sketch. (iii) part-level object detection – detect specified parts (e.g., “head” and “legs” of a “horse”).

#### 3.1. Background

Our framework has two key modules – Object Detection and Sketch-Based Image Retrieval (category-level and fine-grained). For completeness, we give a brief background.

![Figure 2. Faster-RCNN](image)

**Baseline Supervised Object Detection** We briefly introduce a supervised object detection (SOD) framework, Faster-RCNN [67] that remains state-of-the-art [58, 97, 110]. Given a photo \( p \in \mathbb{R}^{H \times W \times 3} \), a backbone feature extractor (VGG [80], or ResNet [33]) \( F_d(\cdot) : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{H' \times W' \times 512} \) computes feature map \( f_p \in \mathbb{R}^{H' \times W' \times 512} \). Next, a two-stage process is followed: (i) Given backbone feature map \( f_p \), a region proposal network (RPN) \( \mathcal{R} : \mathbb{R}^{H' \times W' \times 512} \rightarrow \mathbb{R}^{H \times W \times 3} \) generates rectangular boxes (i.e., proposals) \( r = \{r_1, \ldots, r_R\} \), where \( r \in \mathbb{R}^{R \times 4} \) and “objectness measure” – a scalar [0, 1] probability of the box \( r_j \) having an object. (ii) Using proposals \( r \in \mathbb{R}^{R \times 4} \) we pool the feature map \( f_p \) via RoI pool [26] to get intermediate feature of size \( \mathbb{R}^{7 \times 7 \times 512} \), followed by a fully-connected layer (FC) to get \( f_r \in \mathbb{R}^{R \times 512} \) as, \( f_r = P(f_p, r) \). The patch feature \( f_r \) is branched into two streams – a classification branch \( \phi_{cls} : \mathbb{R}^{R \times 512} \rightarrow \mathbb{R}^{R \times (|C| + 1)} \) outputs probability distribution (per RoI) over \( C \) pre-defined classes and a catch-all background class; a box regressions \( \phi_{reg} : \mathbb{R}^{R \times 512} \rightarrow \mathbb{R}^{R \times 4} \) refines initial box predictions \( r \in \mathbb{R}^{R \times 4} \).

**Baseline SBIR Framework** We recap a baseline SBIR framework. Given a sketch/photo pair \((s, p)\), we use a sketch/photo feature extractor to get the feature map \( f_s = F_d(s) \in \mathbb{R}^{512} \) and \( f_p = F_p(p) \in \mathbb{R}^{512} \). Category-level SBIR requires \((s, p)\) from the same category, whereas fine-grained SBIR requires instance-level sketch/photo matching. For training, the cosine distance \( \delta(\cdot, \cdot) \) to a sketch
anchor (s) from a negative photo \((p^-)\), denoted as \(\beta^- = \delta(f_s, f_p^-)\) should increase while that from the positive photo \((p^+)\), \(\beta^+ = \delta(f_s, f_p^+)\) should decrease. Training is done via triplet loss with hyperparameter \(\mu > 0\),
\[
L_{\text{trip}} = \max\{0, \mu + \beta^+ - \beta^-\} \quad (1)
\]
To extend FG-SBIR across multiple categories (cross-category FG-SBIR), we train with Eq. (1) using “hard-triplets” – \((s, p^+, p^-)\) have same category, and a class discriminator loss across categories using cross-entropy loss,
\[
L_{\text{cat}} = -c_q^i \log \frac{\exp(F_c(f_q^i))}{\sum_{v_j} \exp(F_c(f_q^j))} \quad (2)
\]
where, query \(q = \{s, p\}\), \(c_q^i\) represent class label of \(i^{th}\) sample, \(F_c : \mathbb{R}^{512} \rightarrow \mathbb{R}^{|C|}\) predicts softmax probabilities.

### 3.2. Weakly Supervised Object Detection

![Weakly supervised setup (no bounding box) trains using image-level class labels with classification heads (\(\phi_{cls}\)). Initial prediction \(\omega_0\) is refined in \(K\) steps with \(\phi_{cls}\) to predict \(\omega_k\).](image)

To avoid collecting expensive bounding box annotation, weakly supervised object detection (WSOD) trains using image-level class labels – objects of a class are present or not. To avoid using bounding box annotation, we either use a pre-trained region proposal network\(^1\) (RPN) or heuristic-based selective search [37], or edge boxes [111] that generate box proposals \(r = \{r_1, \ldots, r_N\}\). The patch features \(f_r = \mathcal{P}(f_p, r)\) is branched into a classification head \(x_c = \phi_{cls}(f_r) \in \mathbb{R}^{512\times(|C|+1)}\) and a detection head \(x_d = \phi_{det}(f_r) \in \mathbb{R}^{512\times(|C|+1)}\). The classification head \(\phi_{cls}\) scores individuals proposals into pre-defined \(C\) classes and a catch-all background class via softmax across \((|C| + 1)\) class labels
\[
\sigma_{cls}(x^{i,j}_c) = \frac{\exp(x^{i,j}_c)}{\sum_{k=1}^{C+1} \exp(x^{i,k}_c)} \quad (3)
\]
The detection head \(\phi_{det}\) measures the contribution of each patch \(i (r_i \in r)\) of being classified to class \(j\) in \((C+1)\), (i.e., a patch score for each class) via softmax across \(R\) regions
\[
\sigma_{det}(x^{i,j}_d) = \frac{\exp(x^{i,j}_d)}{\sum_{r=1}^R \exp(x^{i,k}_d)} \quad (4)
\]

\(^1\)Pre-trained [45] RPN is highly generalisable to unseen datasets [29] due to its generic objective that learns to predict “objectness” measure.

We train using image-level labels \(Y = \{y_0, y_1, \ldots, y_{|C|}\}^T \in \mathbb{R}^{(|C|+1)\times 1}\), where \(y_c = 1\) or 0 indicates if instance of class \(c \in C\) is present in the image or not. The combined score (element-wise product) of class score \(\sigma_{cls}\) for each patch and a patch score for each class \(\sigma_{det}\) is computed as, \(\omega_0 = \sigma_{cls}(x_d) \odot \sigma_{det}(x_d)\). Since we only have image-level class labels, from the combined score \(\omega_0 \in \mathbb{R}^{512\times(|C|+1)}\), we take the sum over all patches \(y_i = \sum_{c=1}^{C+1} \omega_{0,c}\) to get the probability of instances from the \(c^{th}\) class present in the image or not. Training happens via multi-class cross entropy,
\[
L_{\text{ws}} = - \sum_{c=1}^{C+1} y_c \log \hat{y}_c + (1 - y_c) \log(1 - \hat{y}_c) \quad (5)
\]

Unlike SOD, using bounding box annotation to refine proposals, WSOD uses only image-level class labels that fail to naively refine proposals. Hence, we use an iterative refinement classifier \(\omega_k = \phi^*_{cls}(f_r)\), where \(\omega_k \in \mathbb{R}^{512\times(|C|+1)}\) to predict a refined class score for each RoI, as shown in Fig. 3. The refinement classifier \(\phi^*_{cls}\) is supervised via pseudo scores labels \(l_{k-1}\) from \((k-1)^{th}\) iteration as, (i) we compute the patches with highest scores in each class \(r^*_c = \text{argmax}_{r_c} (\omega_{c_{k-1}})\). (ii) All regions \(r_i \in r\) that has high overlap with a top scoring patch \(r^*_c\) should be the same class label \(c\) as, \(l^*_c = 1\) if \(\text{IoU}(r_i, r^*_c) \geq 0.5\). (iii) If a region \(r_i \in r\) has low overlap with any top scoring patch \(r^*_c\), we assign it to background class \(l^*_c = 1\). (iv) If a class \(c\) is not in image \(p\) we assign \(l^*_c = 0\). The refinement loss is
\[
L_{\text{ref}}^k = \frac{1}{R} \sum_{r=1}^R \sum_{c=1}^{C+1} \omega_{c_{k-1}}(i,j) \cdot l_{k-1}(i,j) \log \omega_{c_{k-1}}(i,j) \quad (6)
\]

Both SOD and WSOD restrict detection to pre-defined \(C\) classes. In the next section, we overcome this fixed-set limitation using prototype learning with SBIR.

### 3.3. Localising Object Regions with Query Sketch

We replace the fixed-set classifier in WSOD with scalable open-set prototype learning [49]. Each head \(\{\phi_{cls}, \phi_{det}, \phi^*_{cls}\}\) in WSOD that predict scores \(\mathbb{R}^{512 \times 512} \rightarrow \mathbb{R}^{512\times(|C|+1)}\) is modified to compute their respective embedding vectors \(e = \{e_{cls}, e_{det}, e^*_{cls}\}\) as \(\mathbb{R}^{512 \times 512} \rightarrow \mathbb{R}^{512\times 512}\). Next, we compute a support set (prototypes for category-level/instance-level sketch) \(S = [e_{bg}, f_{1}^s, f_{2}^s, \ldots, f_{s}^c]^{T} \in \mathbb{R}^{512\times(|C|+1)}\) by encoding query sketches \(\{s_1, \ldots, s_c\}\) with a pre-trained sketch encoder \(\mathcal{F}_s\) and a learned catch-all background embedding \(e_{bg} \in \mathbb{R}^{512}\), as shown in Fig. 4. The scores \(x, x_d, \omega_k\) (analogous to Sec. 3.2) are computed using \(S\) and embedding vectors \(e\) of detected regions
\[
x_c = e_{cls} \cdot S; \quad x_d = e_{det} \cdot S; \quad \omega_k = e^*_{cls} \cdot S. \quad (7)
\]
Carefully choosing a sketch encoder \(\mathcal{F}_s\) leads to several properties: (i) pre-training \(\mathcal{F}_s\) on category-level SBIR computes \(S\) that detect regions \(r\) with the same category as
query sketches – category-level object detection. (ii) pre-training \( F_s \) on cross-category FG-SBIR computes \( S \) where only instance-level aligned regions \( r \) are detected – fine-grained object detection. (iii) Extending fine-grained object detection with a generalisable (out-of-vocab) sketch encoder \( F_s \) helps to detect object parts (e.g., “head” of a “horse”) given query sketches – part-level object detection.

We train object detection modules \( \{F_d, P, \phi_{cls}, \phi_{det}, \phi_{cls}^s\} \) using Eq. (5) and Eq. (6) in WSOD (Sec. 3.2).

While the sketch encoder \( (F_s) \) trains object detector via prototypes for each category/instance sketch, we further enhance training efficiency with additional supervision from the photo encoder \( (F_P) \), as shown in Fig. 4. Specifically, we impose a \( L1 \)-based feature matching loss (analogous to feature distillation [34]) between patch features \( f_r \) from proposals \( r \) in object detector and the photo feature computed for cropped photo regions \( \text{Crop}(p, r) \) using pre-trained \( F_P \) as, \( \mathcal{L}_{\text{kd}} = ||f_r - F_P(\text{Crop}(p, r))||_1 \). The final loss is,

\[
\mathcal{L}_{\text{tot}} = \mathcal{L}_{\text{ws}} + \sum_{k=1}^{K} \mathcal{L}_{\text{ref}}^k + \lambda ||f_r - F_P(\text{Crop}(p, r))||_1
\]

where the hyperparameter \( \lambda = 1 \). Although, in theory, we can use our baseline SBIR (in Sec. 3.1), training object detection requires learning a generalised (out-of-vocab) SBIR for category-level and fine-grained sketch/photo matching under wide variations like illumination, complex background, occlusions, unseen categories etc.

3.4. Prompt Learning for Generalised SBIR

To train object detection using SBIR with high generalisation and open-vocab capabilities, we introduce prompt learning [107] using CLIP [64] for SBIR (both category-level and cross-category fine-grained). CLIP [64] consists an image and text encoder (e.g., ViT [22], or ResNet [33]) trained on large 400M text/image pairs. This leads to a highly generalisable model that works zero-shot across multiple tasks and datasets. However, adapting CLIP for sketches is tricky since naive fine-tuning leads to model collapse. Hence, we use prompt learning, a set of \( P \) learnable vector \( \nu_s \in \mathbb{R}^{p \times 768} \) for sketch and \( \nu_p \in \mathbb{R}^{p \times 768} \) for photo, injected into the first layer of ViT to induce CLIP to learn downstream sketch/photo distribution. Importantly, prompting CLIP preserves the desired generalisation ability [107] since the knowledge learned by CLIP is distilled into prompt’s weights while keeping the ViT weights frozen. Our new sketch encoder is defined by adapting CLIP’s image encoder using sketch prompt \( (\nu_s) \) as, \( F_s() = F_{\text{clip}}(\cdot, \nu_s) \) and using \( \nu_p \) for photo encoder as, \( F_P() = F_{\text{clip}}(\cdot, \nu_p) \). Since ViT weights are frozen, training our CLIP-based SBIR is parameter-efficient – we train only \( \nu_s \in \mathbb{R}^{p \times 768} \) and \( \nu_p \in \mathbb{R}^{p \times 768} \). This allows training with less data, and faster convergence. For category-level SBIR, \( (\nu_s, \nu_p) \) learns category inducing prompts using triplet loss (in Eq. (1)). Learning cross-category FG-SBIR, is slightly more complicated that trains \( (\nu_s, \nu_p) \) using hard-triplet in Eq. (1), and a modified class discriminative loss Eq. (2) using CLIP’s text encoder as,

\[
\mathcal{L}_{\text{cat}} = -c_q \log \frac{\exp(f_q^t \cdot f_i^s)}{\sum_{j \neq q} \exp(f_q^t \cdot f_j^s)}
\]

where, \( f_i^s \in \mathbb{R}^{512} \) is computed by CLIP’s text encoder as, \( f_i^s = F_{\text{clip}}^{(t)}(\text{“a photo of a } [c_q^i] \text{”}) \) for category \( c_q^i \). Equipped with our novel prompt-based SBIR, we train open-vocab category-level object detection, fine-grained object detection, and part-level object detection.

3.5. Bridging Object-Level and Image-Level

While SBIR is trained using object-level (single object) sketch/photo pairs, object detection works on image-level (multiple objects) data. To train object detectors using SBIR, we need to bridge this object and image-level gap. Our solution is embarrassingly simple – synthesise a canvas of size \( (H \times W) \) by randomly tiling \( n = \{1, \ldots, 7\} \) object-level photos in SBIR datasets [30, 75]. Despite its simplicity, our augmentation trick, analogous to CutMix [101], improves robustness against input corruptions and out-of-distribution generalisation [101, 102]. The paired sketches for photos in canvas are used to construct the support set \( S \). Note, we train our object detector without the need to “see” the evaluation data distribution or use any annotation (bounding box or image-level class labels). We call this setup – extremely weakly supervised object detection (EW-SOD) – no need to “see” the downstream data distribution.
4. Experiments

Dataset  We train our object detector using existing cross-category FG-SBIR dataset – Sketchy [75] that contains 125 categories, each with 100 photos. Every photo in [75] has at least 5 instance-level paired sketches. To evaluate fine-grained object detection, we use SketchyCOCO [25] comprising of natural images in MS-COCO [52] with instance-level paired sketches. Following Liu et al. [53], we select 1,225 sketch/photo pairs from SketchyCOCO [25] with at least one foreground sketched object. We filter the overlapping categories of in SketchyCOCO [25] from Sketchy [75] to measure true zero-shot performance. For category-level object detection, we train on category-level sketch/photo pairs in QuickDraw-Extended [19] having 330k sketches and 204k photos from 110 categories. Following [86], we evaluate on a subset of standard object detection datasets like PASCAL-VOC [23] and MS-COCO [52] that have 20 and 56 overlapping categories in QuickDraw [30].

Implementation Details  Our model is implemented in PyTorch on a 11GB Nvidia RTX 2080-Ti GPU. First, we train a generalised cross-category FG-SBIR with image size $(224 \times 224)$ by adapting CLIP with ViT [22] backbone (ViT-B/32 weights) using prompt learning [39]. The prompts $(P = 3)$ are trained with triplet loss [100], margin $\mu = 0.3$. Adam optimiser with learning rate $\mu$-4 for 60 epochs, and batch size 64. Our object detection pipeline is build using Detectron2 [91]. We use FasterRCNN [67], pretrained on Visual Genome [45] and remove the RoIPooling [26] and subsequent layers to keep only the pretrained backbone ResNet+FPN ($F_d$) [33, 50] and Region Proposal Network ($R$) that generates 1000 proposals. An alternative is to use handcrafted region proposals like selective search [87], but we observed slight performance drop. The object detector trains using SGD with batch size 8 and initial learning rate $5e-3$, multiplied by 0.1 at 150k and 250k iterations. We train in a two-step process: (i) keeping $F_d$ and $R$ fixed, we train the RoI pooling and FC layers $(P)$, classification head ($\phi_{cls}$), detection head ($\phi_{det}$), and refinement head ($\phi_{ref}$) for 240k iterations. (ii) We freeze only $R$ and finetune all modules for 80k iterations. Non-maxima suppression with IoU $\geq 0.3$ is applied to get final predictions.

Evaluation Metric  (i) For fine-grained object detection, we measure $AP_{3}$, $AP_{5}$, and $AP_{7}$ that computes the average precision (AP) at IoU values 0.3, 0.5, and 0.7. (ii) For category-level object detection, we use measure $AP_{5}$ and CorLoc that computes percentage of images for which the most confident predicted box has IoU $\geq 0.5$ with at least one of the ground-truth boxes for every class. (iii) For cross-category FG-SBIR, we measures $Acc. @q$ – percentage of sketches having true matched photo in the top-q list, and (iv) mean average precision (mAP), and precision considering top 200 retrievals P@200 for category-level SBIR.

4.1. Competitors

For object detection, we compare against, (i) supervised object detection (SOD) using both bounding box in addition to sketch/photo annotations: Mod-FRCNN adapts FasterRCNN [67] for unseen class by concatenating query sketch feature with the RoI pooled feature followed by a binary classifier. MatchNet [36] extends Mod-FRCNN using co-attention to generate region proposals conditioned on query sketch along with squeeze-and-co-excitation to adaptively re-weight importance distribution of candidate proposals. CoAttOD [86] improves upon MatchNet by mitigating the sketch/photo domain misalignment using cross-modal attention. (ii) Weakly supervised object detection (WSOD) trains only on image-level sketch annotations without any additional bounding boxes: WSDDN [10] repurposed object detection as a region classification via multiple instance learning (MIL) paradigm. To inject query sketch to WSDDN, we use cross-attention with RoI pooled feature followed by a binary classifier for detection. OICR [84] improves WSDDN with an iterative MIL to refine initial predictions scores to improve discriminatory power for detection. PCL [83] generates multiple positive instance in an image via clustering and assigning proposals to the label of corresponding object class for each cluster. ICMWSD [68] addresses the problem of prior WSOD that focus on the most discriminative part of an object using context information. In particular, ICMWSD obtains a “dropped feature” by dropping the most discriminative parts, followed by maximising the loss of the “dropped feature” that force the network to look in the surrounding context regions. (iii) We adapt <Method> in WSOD to E<<Method> that exclusively training on SBIR datasets [30, 75] by synthesising canvas with randomly tiling $n = \{1, \ldots, 7\}$ object-level photos and using their paired sketches to construct the support $S$. We call this setup – extreme weakly supervised object detection (EWSOD).

For zero-shot category-level SBIR, we compare against: GRL [19] combines similar semantic information (word2vec [56]) of class labels with visual sketch information and trains using a gradient reversal layer [24] to reduce sketch/photo domain gap. VKD [89] is similar to ours using prototype-learning but employ selective knowledge distillation and ViT [22] backbone. For zero-shot cross-category FG-SBIR: CDG is a SOTA domain generalisation method [77] adapted to cross-category FG-SBIR [60] using categories as domain and intra-category sketch/photo pairs as label. CCD [60] models a universal manifold of prototypical visual sketch traits that dynamically embeds sketch/photo, to generalise to unseen categories.

4.2. Generalisability of Cross-Category FG-SBIR

Due to the significant impact of SBIR on training object detectors, it is imperative to learn a powerful cross-
Table 1. Quantitative performance of zero-shot category-level SOD (CL-SBIR) and cross-category FG-SBIR (CC-FGSBIR).

<table>
<thead>
<tr>
<th>Train</th>
<th>CL-SBIR [19]</th>
<th>CC-FGSBIR [75]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mAP</td>
<td>P@200</td>
</tr>
<tr>
<td>GRL</td>
<td>9.0</td>
<td>6.75</td>
</tr>
<tr>
<td>Ours</td>
<td>18.2</td>
<td>36.1</td>
</tr>
<tr>
<td>VKD</td>
<td>15.0</td>
<td>29.8</td>
</tr>
<tr>
<td>Ours</td>
<td>13.1</td>
<td>23.2</td>
</tr>
<tr>
<td>GRL</td>
<td>6.3</td>
<td>5.7</td>
</tr>
<tr>
<td>Ours</td>
<td>9.6</td>
<td>6.3</td>
</tr>
</tbody>
</table>

Figure 6. Qualitative retrieval results for cross-category FG-SBIR.

category FG-SBIR that is highly generalisable. In other words, the accuracy of SBIR puts a bottleneck on object detection performance. Tab. 1 compares category-level SBIR (CL-SBIR) and cross-category FG-SBIR (CC-FGSBIR) on QuickDraw-Extended [19] and Sketchy [75] respectively, using 100%, 70%, and 50% of the training set.

Performance Analysis From Tab. 1 we make the following observations: (i) with decreasing train-set categories, the performance gap (ratio of proposed / SOTA) between the proposed method versus GRL (for CL-SBIR) and CDG (for CC-FGSBIR) increases from 2.1/1.4 at 100% data to 3.0/4.2 at 50% data. This shows the high generalisation potential when using prompt-based CLIP models for sketch/photo matching. (ii) Performance gap of proposed versus SOTAs for 100% → 50% is more significant in CC-FGSBIR as compared to CL-SBIR. Hence, it is more difficult to discriminate unseen intra-category sketch/photo pairs than recognise a novel categories. (iii) Performance of all competitors in CL-SBIR and CC-FGSBIR are staggering inferior to proposed CLIP-based approach. Such a strong SBIR is necessary to unlock training object detection in EWSOD setup (cross-dataset and weakly supervised).

4.3. Category-Level Object Detection

We benchmark on a subset of standard object detection PASCAL-VOC [23] and MS-COCO [52] datasets that have overlapping categories with QuickDraw [30] sketches. Unlike traditional object detection that detects all instances for known classes in an image, category-level object detection specifies the category of interest by drawing a query sketch.

Performance Analysis From Tab. 2 we observe: (i) best SOD method outperform the best WSOD by an average $AP_5$ margin of 1.7%/0.1% in VOC/MS-COCO. This shows that although WSOD performs less than SOD (using additional bounding box annotation), the performance gap is not as significant as generally observed in prior works on seen setup using text as query [29, 68, 84]. In other words, using sketch gives nearly similar performance for zero-shot setup for SOD and WSOD. (ii) EWSOD methods further drops $AP_5$ of best WSOD method by 5.1%/1.6%. This highlights the lack of generalisation of object detectors to the shift in data distribution when trained on SBIR photos and tested on VOC/MS-COCO. (iii) Despite being trained on the challenging EWSOD setup, our proposed method outperforms best SOD by 14.7/10.9, WSOD by 16.4/11.0, and EWSOD by 21.5/13.2 in zero-shot setup. This shows the extreme generalisation potential of training object detection using a strong CLIP-based SBIR.

4.4. Fine-Grained Object Detection

Unlike category-level object detection that detects all instances of sketched category, the goal of fine-grained object detection is to detect only a specific instance for the input query sketch with instance-level alignment.

Performance Analysis From Tab. 3 we observe: (i) Methods in SOD have nearly similar performance as WSOD and drops for EWSOD, similar to that in category-level detection in Tab. 2. (ii) Compared to SOD, the performance of WSOD drops more for $AP_5 \rightarrow AP_7$. This
Table 3. SketchyCOCO detection fine-grained.

<table>
<thead>
<tr>
<th>Method</th>
<th>AP₃</th>
<th>AP₅</th>
<th>AP₇</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mod-FRCNN</td>
<td>2.5</td>
<td>3.5</td>
<td>3.1</td>
</tr>
<tr>
<td>MatchNet</td>
<td>9.3</td>
<td>11.0</td>
<td>10.5</td>
</tr>
<tr>
<td>CoAttOD</td>
<td>10.4</td>
<td>12.1</td>
<td>11.7</td>
</tr>
<tr>
<td>WSOD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WSDDN</td>
<td>8.1</td>
<td>10.2</td>
<td>9.4</td>
</tr>
<tr>
<td>OICR</td>
<td>8.9</td>
<td>10.9</td>
<td>10.0</td>
</tr>
<tr>
<td>PCL</td>
<td>9.2</td>
<td>11.5</td>
<td>10.6</td>
</tr>
<tr>
<td>ICMWSD</td>
<td>10.3</td>
<td>11.9</td>
<td>10.8</td>
</tr>
<tr>
<td>EWSOD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E-WSDDN</td>
<td>6.4</td>
<td>8.5</td>
<td>7.6</td>
</tr>
<tr>
<td>E-OICR</td>
<td>7.1</td>
<td>9.1</td>
<td>8.3</td>
</tr>
<tr>
<td>E-PCL</td>
<td>7.3</td>
<td>9.4</td>
<td>8.7</td>
</tr>
<tr>
<td>E-ICMWSD</td>
<td>8.5</td>
<td>10.2</td>
<td>9.4</td>
</tr>
<tr>
<td>Proposed</td>
<td><strong>15.0</strong></td>
<td><strong>17.1</strong></td>
<td><strong>16.3</strong></td>
</tr>
</tbody>
</table>

Figure 8. Cross-Category Fine-Grained Object Detection using query sketches with images from SketchyCOCO [25].

is since WSOD methods use less accurate selective search [87] and edge boxes [111] for region proposals compared to the more accurate RPN [67] in SOD. (iii) Our proposed method outperforms SOD, WSOD, and EWSOD in zero-shot setup, thereby proving its fine-grained generalisation.

4.5. Part-Level Object Detection

Encouraged with the generalised fine-grained discriminative power of the proposed method in Tab. 3, we go a step further and ask: can we only detect a part (e.g., only ‘head’) of an instance? Due to lack of annotation, a quantitative evaluation of part-level object detection is infeasible. Nonetheless, we conduct a qualitative study by manually editing sketches to create partial sketches of a single part (e.g., only “head” of “horse”). Fig. 9 presents some results (for more see supplementary). We observe that (i) our proposed method can uniquely detect the sketched ‘head’ region of different objects. (ii) Detection performance is lower for ambiguous part sketches like ‘leg’ (e.g., front-leg, back-leg etc.) (iii) Since detection depends on region proposals from RPN, our model fails to detect tiny sketched parts. Tiny object detection [47] is a known challenge for traditional object detection [67].

4.6. Ablation

Selective Search v/s Edge Boxes v/s RPN Unlike the proposed method using pre-trained [45] RPN to generate 1000 box proposals, WSOD methods mostly use selective search [87] (SS) or edge boxes [111] (EB) that do not need pre-training using box annotation from visual genome [45].

Hence, for a fair comparison, we replace RPN with SS/EB drops AP₅ performance by 1.3/2.6 on SketchyCOCO [25].

Influence of Classifier Refinement We observe AP₅ improve by 2.4 and 1.1 for K = 1 → 2 and K = 2 → 3 respectively but a small drop of 0.2 for K = 3 → 4.

Influence of Supervision from Photo Encoder in SBIR Although we can train an object detector using only pretrained sketch encoder (trained on SBIR) via prototype learning, removing supervision from the photo encoder in SBIR drops 4.5 in A₅ on SketchyCOCO [25].

4.7. Limitation and Future Works

Introducing fine-grained object detection using sketch opens several possibilities that we do not consider. Given multiple query sketches, currently we tread them as independent query embeddings. However, a user might be interested in detecting complex scenes (a “dog” on the right of a “person”) with multiple objects that have meaningful spatial alignment. Future works can extend fine-grained object detection to semantic segmentation using complex sketches from the recently introduced FS-COCO [14] dataset.

5. Conclusion

We cultivate the expressiveness that human sketch bring for object detection. The proposed sketch-enabled object detection framework detects what you intend to express in your sketch – an object detector that is both instance-aware and part-aware. Accordingly, we design a novel prompt learning setup to marry CLIP and SBIR, to train a sketch-aware detector, that works without needing bounding box annotation, or class labels. To make our detector general-purpose, we further dictate it to work in a zero-shot manner. While SBIR is trained using object-level (single object) sketch/photo pairs, object works on image-level (multiple categories). We bridge this object and image-level gap using a data augmentation trick that improves robustness towards corruption and generalisation to out-of-vocab. The resulting framework outperforms both supervised, and weakly supervised object detectors on zero-shot setup.

Figure 9. Unlike traditional object detection that detects an entire object (e.g., a “horse”), sketches can express fine-grained RoI to detect a specified part of an object (e.g., the “head” of a “horse”).
References


[23] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. IJCV, 2010. 2, 6, 7


[31] Cusus Ham, Gemma Canet Tarres, Tu Bui, James Hays, Zhe Lin, and John Collomosse. Cogs: Controllable generation and search from sketch and style. In ECCV, 2022. 1, 2


[100] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M. Hospedales, and Chen Change Loy. Sketch me that shoe. In CVPR, 2016. 2, 6

[101] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In ICCV, 2019. 2, 3, 5


[105] Xialin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas Huang. Adversarial complementary learning for weakly supervised object localization. In CVPR, 2018. 3


