SIFTer: Self-improving Synthetic Datasets for Pre-training Classification Models

Anonymous CVPR submission

Paper ID 8

Abstract

As the size of image datasets for pre-training large image classification models grows at a rapid pace, it is becoming increasingly difficult to correct the societal biases in these datasets and mitigate the risks of violating privacy and copyright. Synthetic image datasets that are free of such risks and could be substituted for part of the pre-training. Unlike real image datasets that can only increase in quantity and resolution, the quality of images in synthetic datasets can be improved continuously. However, previous efforts to improve the quality of synthetic datasets have required many trials guided by human intervention. In this study, we attempt to automate the construction of synthetic datasets that achieve high classification accuracy by optimizing a metric (entropy of local features) that correlates with the accuracy on downstream tasks. Using this metric, we constructed HighEnt-1k, a synthetic image dataset that was generated automatically by maximizing the entropy of local features. We applied HighEnt-1k to the pre-training of the DeiT-Tiny model and achieved a classification accuracy of 89.0% in average on 7 fine-tuning datasets. This result is comparable to that of the state-of-the-art VisualAtom model. Furthermore, only a single automated generation trial without any human intervention was needed to achieve this result.

1. Introduction

Billion-scale image datasets such as JFT-4B [29], LAION-5B [28], and IG-3.5B [23] provide researchers the capability to pre-train machine vision backbones at extreme scale. However, as the construction and expansion of datasets for pre-training vision models grow at a rapid rate, it is becoming increasingly difficult to correct for societal biases, and mitigate the risks of violating privacy and copyright. Synthetic image datasets such as FractalDB [15], VisualAtom [30], and Kubric [9] are inherently free of such risks and biases. Unlike real datasets that can only increase in quantity or resolution, the quality of each image in synthetic datasets can be continuously improved.

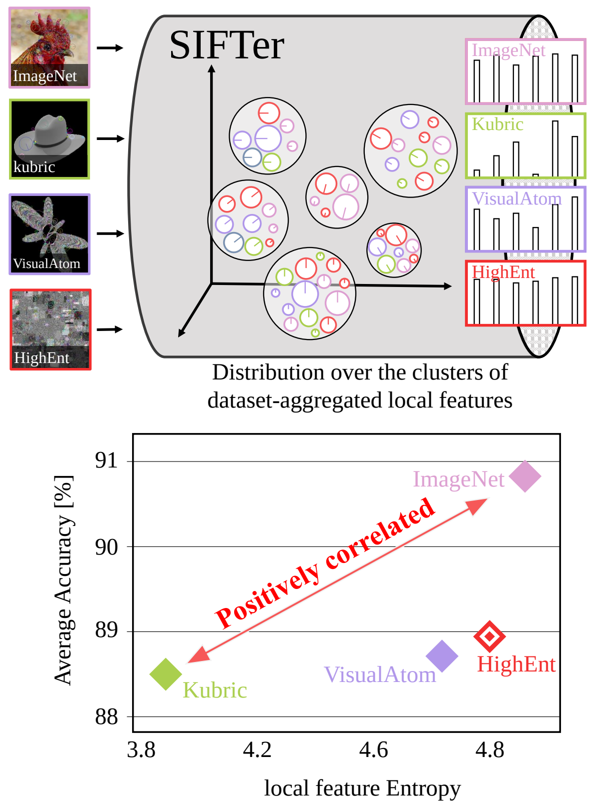


Figure 1. Conceptual diagram of our proposed SIFTer. This technique involves extracting local features from a dataset and calculating feature distributions through k-means clustering. SIFTer uniformly extracts features from datasets, irrespective of whether the images are real or synthetic.

For example, formula-driven supervised learning (FDSL) datasets have been improving continuously from when the fine-tuning accuracy was significantly lower than that of ImageNet-1k [16], to when it became comparable to that of ImageNet-1k [15], then becoming comparable to that of ImageNet-21k [17], and then finally surpassing that of ImageNet-21k [30]. However, without a clear guiding principle as to what comprises a good synthetic image dataset, the process of continuously improving it has taken many years of painstaking human effort. For example, the FractalDB dataset [16] explores the following hyperparameters; the number of categories (8 levels), percentage of fractal region (5 levels), diversity of instances (5 levels), the IFS iterations (4 levels), image size (5 levels). This requires 8 $\times$ 8 $\times$ 2 $\times$ 5 $\times$ 5 $\times$ 4 $\times$ 5 = 64000 trials to cover the entire search space. One pre-training trial takes about 7 hours using 32 GPUs.

In this work, we aim to develop a metric for measuring the pre-training effect of vision datasets, which can be used to continuously improve synthetic datasets without human intervention. By "pre-training effect," we refer to the ability of the dataset to effectively pre-train vision models, so that they show high accuracy on a wide range of downstream tasks on real images and real-world scenarios. Considering the domain gap between real and synthetic datasets, it is surprising that the pre-training effectiveness of some of these synthetic datasets can surpass that of real datasets. We hypothesize that synthetic datasets comprising primitive patterns can still contain rich local features, even though they lack global features with semantic meaning. Preliminary experiments in Section 3.1 show that a model pre-trained on synthetic datasets can tolerate the freezing of the lower layers during fine-tuning (see Fig. 2). This supports our hypothesis that the local features present in the synthetic datasets are indeed useful for training the lower layers. This is also consistent with the results of Yosinski et al. [12], which show that local features acquired in the lower layers of the model during pre-training contribute significantly to the generalizability to downstream tasks.

From these preliminary findings, we choose to focus on the local features to design a metric for measuring the pre-training effect of image datasets. Local features can be extracted from the lower layers of any vision model, but we must first train the model on a specific dataset to obtain such local features. For the purpose of making our local features agnostic to datasets, we have decided to use scale-invariant feature transform (SIFT) features [21, 22]. SIFT features do not require a pre-trained model, and have served a central role in many state-of-the-art (SoTA) image recognition methods [13, 25, 26]. We extract the SIFT features from various real and synthetic datasets and form a codebook using k-means clustering. We then measure the entropy of the distribution of SIFT features from each dataset, and look at the correlation between the entropy and the performance on downstream tasks when a model is pre-trained on that dataset. We observe a positive correlation between the entropy and the performance on downstream tasks. We name our technique for extracting SIFT feature distributions from real and synthetic images "SIFTer".

Using the SIFT feature distributions obtained by SIFTer, we generate a new synthetic dataset that has a high entropy of SIFT feature distributions, which we call HighEnt. We are able to match the pre-training effect of SoTA synthetic datasets with HighEnt. A notable difference between HighEnt and previous synthetic datasets is that HighEnt is able to achieve a SoTA pre-training effect on its first attempt, while other datasets require many months or years of trial and error to achieve the same pre-training effect. The generation process of HighEnt is entirely automated, and the entropy of SIFTer is used as a target metric.

We acknowledge that local features alone cannot fully explain the strong zero-shot downstream performance of pre-trained vision models. Co-occurrence of these local features and their composition as a hierarchy of semantic labels are currently missing from our synthetic datasets. However, as a first step towards automating the process of continuously improving synthetic datasets, we believe that our findings are encouraging. We envision that risks and biases in large real image datasets can be partially alleviated by substituting parts of the pre-training with synthetic images. By automating the process of continuously improving synthetic datasets, we hope to accelerate the development in this direction.

Our main contributions can be summarized as follows:

- We proposed a pipeline SIFTer to compute statistical indicators based on SIFT local features of image datasets.
- Using SIFTer, we measured the entropy of local feature distributions and found that they were correlated with the accuracy in downstream tasks.
- Using the entropy of SIFT feature distributions, we generated a new synthetic dataset HighEnt that maximizes this metric, and showed the SoTA performance partially in among synthetic dataset for pre-training the vision transformers.

2. Related work

2.1. Synthetic datasets

Synthetic datasets in computer vision can be broadly categorized into those that try to simulate real images, and those that comprise primitive patterns that do not resemble any physical objects. The following are examples of the former category. Hataya et al. [10] proposed an extension of ImageNet using datasets generated by the Stable Diffusion model. Frid-Adar et al. [7] constructed a dataset using a generative adversarial network for liver lesion classification. Hinterstoisser et al. [11] constructed a synthetic dataset for object detection, in which they used background images with realistic shapes and texture on top of which they rendered the objects of interest. Chen et al. [3] proposed a synthetic dataset for semantic segmentation, which was validated on Virtual KITTI to KITTI, and from SYNTHIA to Cityscapes. Ward et al. [32] produced a dataset for leaf instance segmentation that contained synthetic images of plants inspired by domain randomization. CLEVR by Johnson et al. [14] is a synthetic dataset comprising sim-

CVPR #8

ple 3D shapes that are used for visual reasoning. ScanNet by Dai et al. [4] is an RGB-D video dataset containing 2.5 million views from more than 1500 scans of indoor scenes. SYNTHIA by Ros et al. [27] is a collection of synthetic images with semantic labels of urban scenes. Virtual KITTI by Gaidon et al. [8] is a synthetic video dataset with accurate ground truth for object detection, tracking, scene and instance segmentation, depth, and optical flow. These synthetic datasets are all designed for a specific purpose, and are not aimed towards learning general visual representations. Kubric [9] was developed to address these shortcomings by generating photo-realistic synthetic datasets for myriad vision tasks, with fine-grained control over data complexity and rich ground truth annotations.

2.2. Formula-driven supervised learning (FDSL)

Unlike the synthetic datasets described in the previous subsection, which simulate real images, FDSL [15] and Shader [1] provide a rich variety of primitive patterns that facilitate the learning of visual representations. In particular, FDSL exploits the unique property of synthetic datasets, which is that the quality of the dataset can be continuously improved. Each reincarnation of FDSL – FractalDB [15], ExFractalDB [18], RCDB [17], and VisualAtom [30] – has shown a monotonic increase in the downstream accuracy. However, these empirical improvements still lack a theoretical explanation as to why each FDSL generation shows better performance.

2.3. Understanding the pre-training effect of FDSL

Yosinski et al. [12] showed that local features acquired in the lower layers of the model during pre-training contribute significantly to its generalizability to downstream tasks. Previous experiments on FDSL [15] also showed that freezing the lower layers did not result in any accuracy degradation during fine-tuning. This leads us to believe that the coverage of local features plays an important role in the downstream performance when pre-training with FDSL.

3. Method

To construct an FDSL dataset with highly effective pre-training in a small number of exploratory experiments, we need an indicator that does not require training. To achieve this goal, we investigate the factors present in both real and synthetic images that lead to effective pre-training. First, we investigate which parts of the vision transformer (ViT) model are affected by pre-training based on layer freezing experiments. Next, based on the results obtained from the layer freezing experiments, we design SIFTer, a pipeline that extracts the distribution of local features acquired in the lower layers, which is the factor that improves the pre-training effect. Then, using the distribution obtained by SIFTer, we investigate what values are correlated with the performance on the downstream tasks. Finally, we construct a dataset HighEnt, which is expected to have highly effective pre-training, by SIFTer-based evaluation.

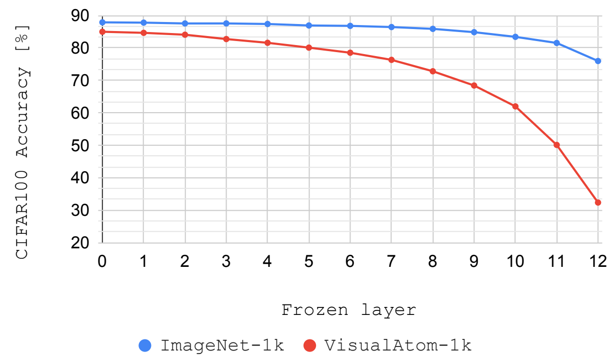


Figure 2. Transition in identification accuracy when freezing the updates during fine-tuning from low to high layers; VisualAtom-1k contains fewer real-world objects than ImageNet-1k, but maintains comparable identification accuracy when frozen at lower layers.

3.1. What layers are important in pre-training?

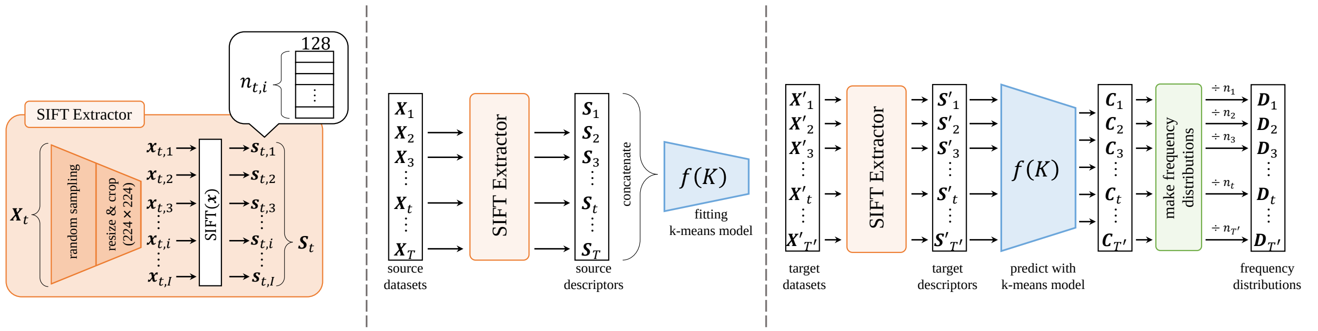
We confirm the importance of primitive features acquired in the lower layers in ViT pre-training through experiments in which the parameter updates in each layer are fixed (frozen) during fine-tuning. We use the DeiT-Tiny model [31] pre-trained on VisualAtom-1k [30] and ImageNet-1k [5] during experiments, and we freeze the 12 transformer blocks in the model step by step, starting from the lowest layer. Then we investigate which blocks contribute to improving the accuracy of the downstream task of image classification using CIFAR100. The experimental results are shown in Fig. 2. The results show that the performance loss during fine-tuning is negligible when layers close to the input are frozen, regardless of whether VisualAtom or ImageNet1k is used for pre-training. Therefore, we conclude that pre-training optimizes the low-layer weights of the ViT model for both synthetic and real image datasets.

3.2. SIFTer

Based on the results of Section 3.1, we focus on the local features of the image as an indicator to explain the pre-training effect. There are several methods for extracting local features from images, but we use SIFT because (1) it can be applied regardless of the real or synthetic image domain and (2) it is a reliable method that has been widely used in past SoTA methods.

3.2.1 Extracting local features from the dataset

Algorithm 1 is used to extract SIFT features from a given dataset $\boldsymbol{X}_t$. The sampled image is resized to 224×224



(a) Extract SIFT from a dataset (b) Clustering SIFT features (c) Examine SIFT frequency distributions

Figure 3. Procedure for extracting local feature frequency distributions from a dataset using SIFTer. (a) Sampling SIFT features extracted from image dataset $\boldsymbol{X_t}$. (b) Constructing a k-means model f that classifies SIFT features into K clusters using the datasets $\boldsymbol{X_1}, \boldsymbol{X_2}, \ldots, \boldsymbol{X_T}$ for codebook creation. (c) Using the constructed k-means model f to classify SIFT features from the dataset $\boldsymbol{X'_t}$ to be analyzed into clusters and to construct the frequency distribution of local features.

Algorithm 1 Extracting SIFT features from a dataset

Require: $I \in \mathbb{N}$
procedure EXTRACT_SIFT($\boldsymbol{X_t}, I$)
  $\boldsymbol{X_{t(\text{sampled})}} \leftarrow$ random_sampling($\boldsymbol{X_t}, I$)
  for $i = 1$ to I **do**
    $\boldsymbol{x_{t,i}} \leftarrow$ resized_crop($\boldsymbol{X_{t(\text{sampled})}}[i]$)
    $\boldsymbol{s_{t,i}} \leftarrow$ SIFT($\boldsymbol{x_{t,i}}$) ▷ extract SIFT from a $\boldsymbol{x_{t,i}}$
  end for
  $\boldsymbol{S_t} \leftarrow$ concat($\boldsymbol{s_{t,1}}, \boldsymbol{s_{t,2}}, \cdots, \boldsymbol{s_{t,I}}$)
  return $\boldsymbol{S_t}$
end procedure

Algorithm 2 Clustering SIFT features

Require: $T, K, I \in \mathbb{N}, , f$: k-means model
for $t = 1$ to T **do**
  $\boldsymbol{S_t} \leftarrow$ EXTRACT_SIFT($\boldsymbol{X_t}, I$) ▷ extract SIFT
end for
$\boldsymbol{S_{all}} \leftarrow$ concat($\boldsymbol{S_1}, \boldsymbol{S_2}, \cdots, \boldsymbol{S_T}$)
$f.fit(\boldsymbol{S_{all}}, K)$ ▷ fitting f with K clusters

Algorithm 3 Examine SIFT frequency distribution

Require: $T', K, I \in \mathbb{N}, f$: k-means model (fitted)
for $t = 1$ to T' **do**
  $\boldsymbol{S'_t} \leftarrow$ EXTRACT_SIFT($\boldsymbol{X'_t}, I$) ▷ extract SIFT
  $\boldsymbol{C_t} \leftarrow f.pred(\boldsymbol{S'_t})$ ▷ predict clusters
  $\boldsymbol{D_t} \leftarrow \boldsymbol{0} \in \mathbb{R}^K$
  for all $c \leftarrow \boldsymbol{C_t}$ **do** ▷ count SIFT for each clusters
    $\boldsymbol{D_t}[c] \leftarrow \boldsymbol{D_t}[c] + 1$
  end for
  $n_t \leftarrow$ len($\boldsymbol{S'_t}$)
  for $c = 1$ to K **do**
    $\boldsymbol{D_t}[c] \leftarrow \boldsymbol{D_t}[c] / n_t$
  end for
end for
return $\{\boldsymbol{D_t}\}_{t=1}^{T'}$

pixels as in training. This prevents the number of SIFT features from changing due to differences in image size, thus allowing a fair comparison.

3.2.2 Clustering of SIFT features

The obtained SIFT features are assigned to exist near a particular codebook (cluster). In this study, we use the k-means model to determine cluster identities. The k-means model is pre-trained using SIFT features from diverse datasets. Algorithm 2 is used for training the k-means model. The k-means model can assign a given SIFT feature to one of K clusters $\{c_1, c_2, \ldots, c_K\}$ of SIFT features. Let $f.fit(\boldsymbol{S}, K)$ be the operation of training a k-means model f so that it can classify features into K clusters using N SIFT features $\boldsymbol{S} = \{\boldsymbol{s_i}\}_{i=1}^{n}$. The operation of predicting and mapping a d-dimensional feature $\boldsymbol{s} \in \boldsymbol{R}^d$ to the kth cluster c_k by the k-means model f is represented as the mapping transformation $f.pred : \mathbb{R}^d \to \mathbb{N}$.

3.2.3 Obtaining the local feature frequency distribution by SIFTer

SIFT feature $\{\boldsymbol{S'_1}, \boldsymbol{S'_2}, \ldots, \boldsymbol{S'_{T'}}\}$ is extracted from each dataset $\{\boldsymbol{X'_1}, \boldsymbol{X'_2}, \ldots, \boldsymbol{X'_{T'}}\}$ to be analyzed, and each $\boldsymbol{S'_1}$ is transformed into a sequence of SIFT cluster numbers $\boldsymbol{C_1}, \boldsymbol{C_2}, \ldots, \boldsymbol{C_{T'}}$ by the k-means model f. The SIFT feature frequency distribution $\boldsymbol{D_t}$ for each cluster is then computed by accounting for the number of SIFT features belonging to each cluster and finally dividing and normalizing the result by the total number n_t of SIFT features extracted from $\boldsymbol{X'_t}$.

3.2.4 Search for indicators using local feature frequency distributions

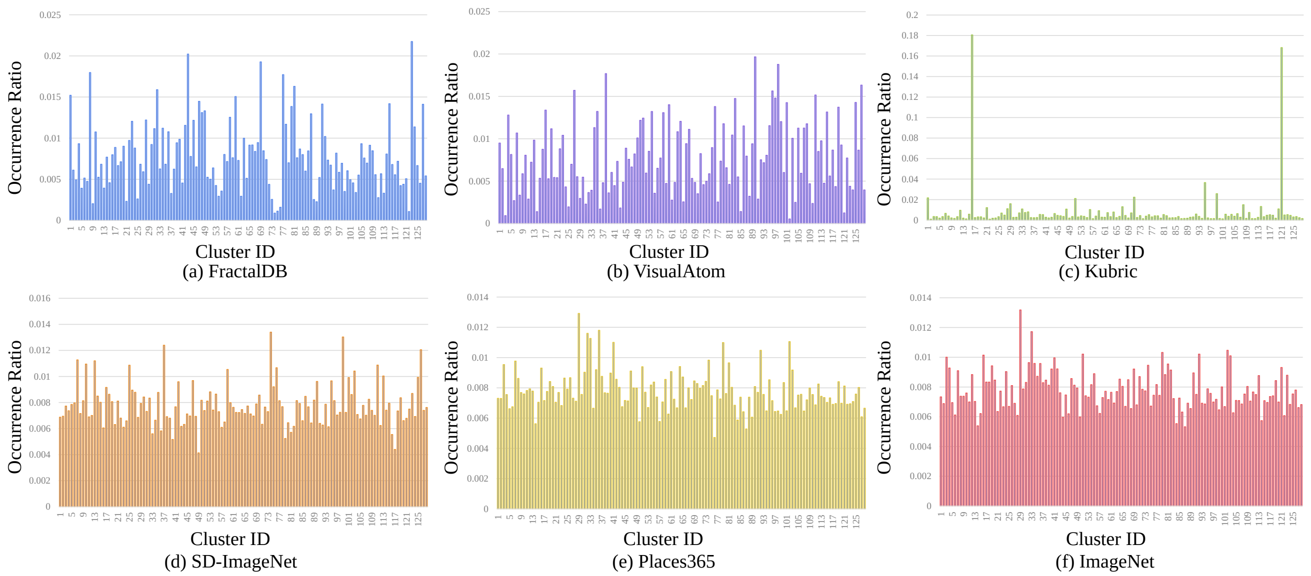


Figure 4. Local feature frequency distributions for multiple real and synthetic image datasets obtained using SIFTer. Datasets with real images or similar domains (d,e,f) contain a similar proportion of clusters, while other datasets (a,b,c) show variations in the frequency of clusters.

The correlation between the distribution of local features obtained by SIFTer and the accuracy of classification is calculated. However, since it is unclear what "correlates" with the pre-training effect of the distribution obtained by SIFTer, we visualize the actual distribution obtained. Here, Figure 4 shows the local feature distribution for each conventional dataset obtained using SIFTer. From figure 4, the feature distributions of (a) FractalDB, (b) VisualAtom, and (c) Kubric, which use only FDSL and 3D models, are biased. On the other hand, the feature distributions of (d) SD-ImageNet, (e) Places365, and (f) ImageNet, which use real images or are generated based on real images, are broad. From the Table 1, the latter group performed better than the former group in this experiment, thus we claim that it is important to cover a wide variety of local features equally. To confirm this, we will conduct an investigation on more detailed distribution indices in 4.3.

Based on our observations of feature frequency distributions computed by SIFTer, we hypothesize that the distributions obtained in the pre-training dataset that are not over- or under-distributed, or similar to the target task, enhance the performance of the downstream task.

To demonstrate this, in this experiment, three statistics from the SIFT feature frequency distribution $\{\boldsymbol{D_t}\}_{t=1}^{T}$ measured on the dataset $\boldsymbol{X_t}$ to be analyzed are evaluated: 1) entropy, 2) Kullback-Leibler divergence (KLD) for the target task, and 3) recall coverage for the target task. These were compared and evaluated as indicators of the coverage of local features. We use ImageNet100 as our target task with a representative real image dataset.

(1) Entropy: The entropy $H(\boldsymbol{D_t})$ at $\boldsymbol{D_t}$ for each dataset $\boldsymbol{X'_t}$ is calculated by the following formula:

$$H(\boldsymbol{D_t}) = -\sum_{c=1}^{K} \boldsymbol{D_t}[c] \ln \boldsymbol{D_t}[c]. \tag{1}$$

By evaluating entropy $H(\overline{\boldsymbol{D}}_{\boldsymbol{t}})$, we can estimate whether each dataset $\boldsymbol{X_t}$ contains an equal amount of diverse features.

(2) Kullback-Leibler divergence: We evaluated the KL distance in the SIFT feature frequency distribution between each pre-trained dataset $\boldsymbol{X'_t}$ and the target task. If $\boldsymbol{D_{IN100}}$ is the SIFT feature frequency distribution of ImageNet_100, the KL distance $KL(\boldsymbol{D_t}||\boldsymbol{D_{IN100}})$ from $\boldsymbol{D_t}$ can be calculated as follows:

$$KL(\boldsymbol{D_t}||\boldsymbol{D_{IN100}}) = \sum_{c=1}^{K} \boldsymbol{D_t}[c] \ln \frac{\boldsymbol{D_t}[c]}{\boldsymbol{D_{IN100}}[c]}. \tag{2}$$

By calculating $KL(\boldsymbol{D_t}||\boldsymbol{D_{IN100}})$, it is possible to quantify how close the distribution obtained by SIFTer is to that of IN_100.

(3) Recall: In addition, to more directly evaluate the coverage of local features quantitatively, we also evaluated the percentage of clusters in which SIFT features in the target task appear at least once, also appear at least once in X_t. This ratio is called recall to the target task. Recall $R(\boldsymbol{D_t}|\boldsymbol{D_{IN100}})$ for the target pre-training dataset $\boldsymbol{X_t}$ is calculated by the following formula:

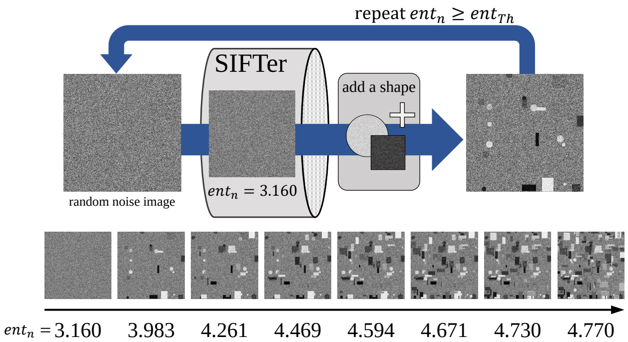


Figure 5. Conceptual diagram of the HighEnt generation method. An image with random noise is used as the initial input, and geometric shapes are added until the entropy of local features exceeds ent_th. The added shapes are circles and rectangles, each of which contains noise.

$$R(\boldsymbol{D_t}|\boldsymbol{D_{IN100}}) = \frac{\sum_{c=1}^{K} u(\boldsymbol{D_t}[c])u(\boldsymbol{D_{IN100}}[c])}{\sum_{c=1}^{K} u(\boldsymbol{D_{IN100}}[c])}, \quad (3)$$

$$u(t) = \begin{cases} 1 & \text{if } t > 0 \\ 0 & \text{if } t = 0. \end{cases} \quad (4)$$

3.3. HighEnt

3.3.1 Generation methodology

As discussed in later sections, the pre-training performance is positively correlated with the entropy of the distribution obtained by SIFTer. Taking advantage of this, we propose a pre-training dataset HighEnt consisting of only images with high entropy: HighEnt is a dataset of 1,000 instances of 1,000 classes, consisting of 1 million images in total. The generation of HighEnt is shown in Figure 5.

The generation procedure of HighEnt consists of two stages. In the first stage, a SIFTer is used to generate a number of images of the class (i.e. 1,000) whose entropy exceeds a certain threshold value. In the second stage, 1,000 instances are generated from one image by splitting the image into patches and shuffling the patches. By putting the instances generated by the same image into the same class, a dataset of 1,000 instances of 1,000 classes is generated. To generate high-entropy images in the first stage, we adopt an annealing-based optimisation. A schematic diagram of the generation algorithm is shown in Figure 5. The initial image is a monochrome random noise image and at each step, a noisy figure (circle/square) is drawn at a random position in the image. The position, size, color and noise intensity of the figure are randomly selected from a pre-determined range. The entropy of the resulting image is measured using SIFTer and the difference $ent_{\text{diff}} = ent_n - ent_{n-1}$ between the entropy before drawing is calculated. If ent_{diff} is positive (i.e. entropy has increased due to drawing), then the change is reflected, otherwise the change is rejected with a probability depending on the magnitude of ent_{diff} and the current number of steps. This operation is repeated until the entropy exceeds ent_{Th}, a threshold value. In the present study, a threshold value of approximately 4.77 was used. This is the maximum value of entropy obtained when the distribution is calculated using SIFTer for each image in ImageNet1k. This method produces 1,000 high-entropy images. In the second stage, the 1,000 images obtained in the first stage are used to generate 1,000 instances from each image. The distribution of local features is considered to be somewhat robust to the operation of swapping image regions. " Based on this, the images were divided into 16 patches of 4x4 each and instances were generated by swapping the order of each patch. We generated 1,000 permutations, different for each image, and used these permutations to perform the above method, generating 1,000 instances for each image. The detailed algorithm is described in the supplementary material.

3.3.2 Preliminary experiments

HighEnt is generated to maximize the entropy of each image, but also to check whether the dataset of these images has an overall high entropy. Thus, 100,000 images are randomly sampled from HighEnt and the entropy of the distribution obtained by SIFTer is calculated. The result is 4.787, which is a very high value compared to other FDSL datasets, and the HighEnt created in this study has a high entropy for the dataset as a whole.

4. Experiments

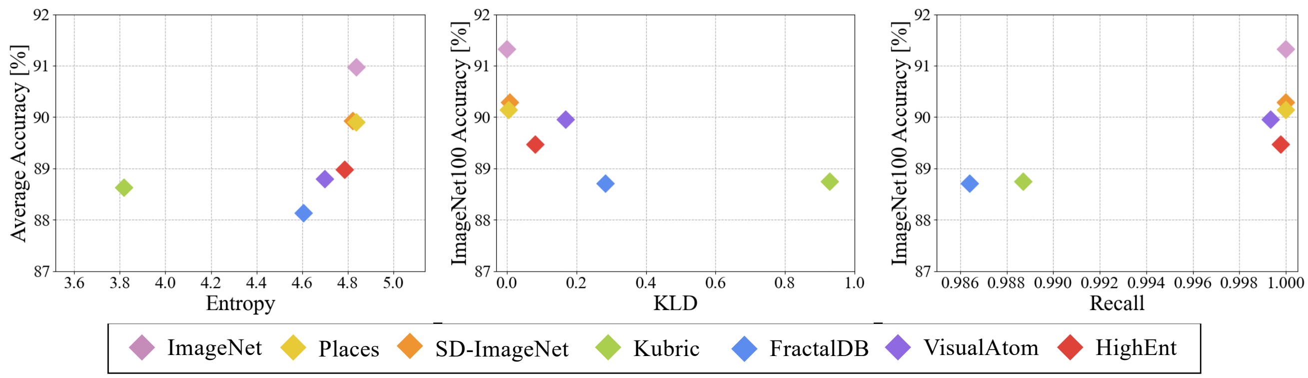
4.1. Experimental setup using the SIFT feature frequency distribution

Dataset used for clustering: We conduct our experiments on a wide range of image datasets than can be largely categorized into three: "real image data" (real), "synthetic image data that partially use real images" (semi-synthetic), and "synthetic image data synthesized by mathematical formulas" (synthetic). We sampled 720,000 images from the following 8 datasets to include images from these categories as evenly as possible: ImageNet1k [5]—general-purpose real image dataset; ADE-20k [34]—real image dataset suitable for scene analysis; SD-ImageNet1k [10]—ImageNet-like dataset with stable diffusion; Kubric—synthetic image dataset generated by capturing ShapeNet [2]; and two formula-based synthetic datasets, VisualAtom and FractalDB. We use these selected images to perform local feature clustering. Details of the image sampling are provided in the supplementary material.

Datasets for analysis: We perform our analysis using multiple datasets as well. Specifically, we use ImageNet and

Table 1. Comparison with fine-tuning accuracy on 7 real image datasets. Experimental results show the highest accuracy for each real/semi-synthetic/synthetic image framework in bold.

Pre-training Dataset	Image	C10	C100	Cars	Flowers	VOC12	P30	IN100	Average
Scratch	–	80.9	64.4	13.6	73.9	56.4	76.9	74.8	63.0
ImageNet1k	Real	**98.4**	**87.9**	**90.2**	**99.0**	**87.9**	82.1	**91.3**	**91.0**
Places365	Real	97.9	85.5	89.5	98.9	84.7	**82.7**	90.1	89.9
SD-ImageNet1k	Semi-Synthetic	**98.1**	**85.7**	**89.3**	**99.0**	**85.3**	**81.8**	**90.3**	**89.9**
Kubric	Semi-Synthetic	97.7	84.5	87.3	98.3	82.9	81.0	88.7	88.6
FractalDB	Synthetic	96.8	81.6	86.0	98.3	80.6	78.4	88.3	87.1
VisualAtom	Synthetic	97.6	84.9	88.8	**98.9**	82.0	81.2	**90.3**	**89.1**
HighEnt (ours)	Synthetic	**97.4**	**85.0**	**89.1**	98.5	**82.0**	**81.4**	89.5	89.0



(a) SIFT Entropy ($r = 0.501$)

(b) KLD from ImageNet100 ($r = -0.699$)

(c) Recall for ImageNet100 ($r = 0.804$)

Figure 6. **Relationship between three statistics of SIFT frequency distribution $\boldsymbol{D_t}$ and fine-tuning accuracy for each pre-training dataset $\boldsymbol{X_t}$.** (a) SIFT entropy: $H(\boldsymbol{D_t})$; (b) KL divergence from ImageNet_100: $KL(\boldsymbol{D_t}||\boldsymbol{D_{IN_{100}}})$; and (c) Recall for ImageNet_100: $R(\boldsymbol{D_t}|\boldsymbol{D_{IN_{100}}})$. r is the correlation coefficient between average/ImageNet_100 fine-tuning accuracy and each statistic.

Places365 [33]—real image datasets consisting of diverse place images; SD-ImageNet and Kubric as semi-synthetic image datasets that use real images; and FractalDB and VisualAtom as synthetic datasets that do not have any real images. We also include our HighEnt for comparison.

Hyperparameters: In this experiment, the number of images I to be sampled from the dataset is set to $100,000$. The number of clusters K is set to 128 for entropy and KLD evaluation for the target task, and $32,768$ for recall ealuation for the target task. Details of the hyperparameter search are described in the supplementary material.

4.2. Local feature distribution obtained by SIFTer

Figure 4 shows the local feature distribution for each conventional dataset obtained using SIFTer. It is observed that datasets of (a) FractalDB, (b) VisualAtom and (c) Kubric have significant gaps in densities, while datasets of (d) SD-ImageNet, (e) Places365 and (f) ImageNet have less gap. Table 1 shows the summary of performance on various downstream classification tasks. From the table, (d) SD-ImageNet, (e) Places365 and (f) ImageNet tend to perform better than (a) FractalDB, (b) VisualAtom and (c) Kubric. This observation suggests that having less gaps in the density over the local feature space would be beneficial for downstream tasks. To clarify this, we conduct further investigation on different distribution metrics in Section 4.3.

4.3. Relationship between the SIFT feature frequency distribution and pre-training effect

We measure and compare each of the three statistics: entropy of the SIFT feature frequency distribution, KL distance to the target task, and recall with the target task. In this experiment, we use ImageNet100 as the target task.

To evaluate the pre-training effect of each dataset, we also measure the fine-tuning accuracy of the classification task on several real image datasets after pre-training ViT-Tiny on the classification task.

We use a total of seven real image datasets for fine-tuning: CIFAR10 (C10) [20], CIFAR100 (C100) [20], Stanford Cars (Cars) [19], Stanford Flowers (Flowers) [24], Pascal VOC 2012 (VOC12) [6], Places 30 (P30) [15], and ImageNet-100 (IN100) [15]. Hyperparameters for ViT pre-

Table 2. Comparison of with fine-tuning accuracy on 7 real image datasets. Each row represents the HighEnt results generated using different Ent_{Th}.

ent_{Th}	Entropy	C10	C100	Cars	Flowers	VOC12	P30	IN100	Average
3.0	3.3	96.0	81.6	80.1	97.9	77.9	81.5	88.7	86.3
3.5	3.8	97.3	84.6	87.9	98.5	80.7	81.3	89.6	88.6
4.0	4.3	97.2	84.0	87.0	98.4	80.1	81.3	88.9	88.1
4.5	4.6	97.3	84.5	87.0	98.6	81.4	81.5	89.6	88.7

training, fine-tuning and data augmentation are adopted from the previous study [17] More detailed settings are described in the supplementary material. Again, the performance of each pre-trained dataset for each target task is as summarized in Table 1, and scatter plots showing the relationship between the statistics of each SIFT feature frequency distribution and fine-tuning accuracy, as well as the distribution statistics for each dataset, are shown in Fig 6.

First, we observe a positive correlation between the entropy of the SIFT feature frequency distribution and the average accuracy of fine-tuning (Fig. 6(a)). Higher entropy implies that the dataset contains a greater variety of local features and has a wider range of local features that can be learned. The result shows a tendency in which higher entropy results in higher average accuracy; thus, we believe this provides an evidence that supports aforementioned hypothesis.

A strong negative correlation was observed between the ImageNet100 accuracy and the KLD between the ImageNet100 SIFT distribution and that of the pre-trained dataset. This suggests that even if the entropy of the frequency distribution is high, the discrimination accuracy tends to deteriorate when the distribution patterns are far from each other in the KLD metric.

A strong positive correlation is observed between the mean accuracy of fine-tuning and recall of the SIFT feature frequency distribution. This suggests that the coverage of local features of the target task is important. However, in some of the pre-training datasets, the accuracy of the fine-tuning varies despite the fact that the recall for the target task is 1.0; that is, all the local features of the target task are covered. This indicates that the recall for the target task alone does not fully explain the pre-training effect.

4.4. Performance evaluation of HighEnt

The accuracy of the pre-training model with HighEnt created for each downstream task is shown in the bottom row of Table 1. As a comparison, the results are shown for a pre-training model using VisualAtom, which is currently the FDSL dataset with the best performance. This result confirms that HighEnt slightly outperforms VisualAtom in the present experimental setup. The average accuracy reported in the VisualAtom paper is 89.1, which is very close to that of HighEnt obtained in this study. While VisualAtom performed multiple searches for the parameters used in its generation, the HighEnt generated in this study achieved comparable performance without such searches. This result suggests the possibility of automating the composition of high-performance FDSL datasets by means of a search based on the indices obtained by SIFTer. Table 2 shows the results of generating and evaluating HighEnt using different values of ent_{Th} used for generation. Although average performance generally improves as entropy increases, it can also be seen that performance is higher when ent_{Th} is 3.5 than when ent_{Th} is 4.0. This indicates that entropy is only a rough indicator of performance.

4.5. Limitations

Although the entropy in the feature distribution of the dataset obtained by SIFTer correlates with the performance on downstream tasks, there are also outlier datasets such as Kubric. Our findings are only an approximation, and not an indicator that can precisely predict pre-training performance. We used circles and rectangles as the shapes for generating HighEnt, but other shapes and objects can be used, and there is room for investigation as to which shape has a high entropy and high pre-training effect. The augmentation method used in the creation of HighEnt has not been investigated using other methods.

5. Conclusion

In this study, a synthetic image dataset with high pre-training effect was automatically constructed with fewer exploratory experiments. Based on preliminary experiments, we focused on local features of images and proposed SIFTer, a pipeline for extracting local features. Using SIFTer, we found a correlation between the entropy of the obtained distribution and the pre-training effect, and developed HighEnt, an image data set generated to have high entropy. HighEnt achieved the same level of accuracy as the conventional SoTA dataset VisualAtom without any parameter search. While the findings of this study do not reveal all of the factors that make pre-training effective, they do suggest that indicator-based data improvement can automatically generate datasets that are effective.

References

[1] Manel Baradad, Chun-Fu Chen, Jonas Wulff, Tongzhou Wang, Rogerio Feris, Antonio Torralba, and Phillip Isola. Procedural image programs for representation learning. In *Advances in Neural Information Processing Systems*, 2022. 3

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 6

[3] Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1841–1850, 2019. 2

[4] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 3

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3, 6

[6] Mark Everingham, SM Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes Challenge: A Retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 7

[7] Maayan Frid-Adar, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Synthetic data augmentation using gan for improved liver lesion classification. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 289–293, 2018. 2

[8] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4340–4349, Los Alamitos, CA, USA, 2016. IEEE Computer Society. 3

[9] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: a scalable dataset generator. 2022. 1, 3

[10] Ryuichiro Hataya, Han Bao, and Hiromi Arai. Will Large-scale Generative Models Corrupt Future Datasets? In *ICCV*, 2023. 2, 6

[11] Stefan Hinterstoisser, Olivier Pauly, Hauke Heibel, Martina Marek, and Martin Bokeloh. An annotation saved is an annotation earned: Using fully synthetic training for object instance detection. In *In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2019. 2

[12] Yoshua Bengio Hod Lipson Jason Yosinski, Jeff Clune. How Transferable are Features in Deep Neural Networks? In *Advances in neural information processing systems*, 2014. 2, 3

[13] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3304–3311, 2010. 2

[14] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1988–1997, 2016. 2

[15] Hirokatsu Kataoka, Kazushige Okayasu, Asato Matsumoto, Eisuke Yamagata, Ryosuke Yamada, Nakamasa Inoue, Akio Nakamura, and Yutaka Satoh. Pre-training without Natural Images. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2020. 1, 3, 7

[16] Hirokatsu Kataoka, Asato Matsumoto, Ryosuke Yamada, Yutaka Satoh, Eisuke Yamagata, and Nakamasa Inoue. Formula-driven Supervised Learning with Recursive Tiling Patterns. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4098–4105, 2021. 1, 2

[17] Hirokatsu Kataoka, Ryo Hayamizu, Ryosuke Yamada, Kodai Nakashima, Sora Takashima, Xinyu Zhang, Edgar Josafat Martinez-Noriega, Nakamasa Inoue, and Rio Yokota. Replacing Labeled Real-Image Datasets With Auto-Generated Contours. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21232–21241, 2022. 1, 3, 8

[18] Hirokatsu Kataoka, Kazushige Okayasu, Asato Matsumoto, Eisuke Yamagata, Ryosuke Yamada, Nakamasa Inoue, Akio Nakamura, and Yutaka Satoh. Pre-training without Natural Images. *International Journal of Computer Vision (IJCV)*, 130(2):990–1007, 2022. 3

[19] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D Object Representations for Fine-grained Categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 7

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning Multiple Layers of Features from Tiny Images. 2009. 7

[21] David G. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1150–1157, 1999. 2

[22] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004. 2

[23] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the Limits of

Weakly Supervised Pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 181–196, 2018. 1

[24] Maria-Elena Nilsback and Andrew Zisserman. Automated Flower Classification over a Large Number of Classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008. 7

[25] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pages 143–156. Springer, 2010. 2

[26] James Philbin, Ondřej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 2

[27] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, 2016. 3

[28] Christoph Schuhmann, Romain Beaumont, Cade W Gordon, Ross Wightman, mehdi cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Mitchell Wortsman, Richard Vencu, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An Open Large-Scale Dataset for Training Next Generation Image-Text Models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 1

[29] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 1

[30] Sora Takashima, Ryo Hayamizu, Nakamasa Inoue, Hirokatsu Kataoka, and Rio Yokota. Visual atoms: Pre-training vision transformers with sinusoidal waves. In *Conference on Computer Vision and Pattern Recognition 2023*, 2023. 1, 3

[31] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training Data-efficient Image Transformers & Distillation through Attention. In *International Conference on Machine Learning*, pages 10347–10357, 2021. 3

[32] Daniel Ward, Peyman Moghadam, and Nicolas Hudson. Deep leaf segmentation using synthetic data. In *CVPPP Workshop at BMVC*, 2018. 2

[33] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 7

[34] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6