Abstract—This work addresses scaling up the sketch classification task into a large number of categories. Collecting sketches for training is slow and tedious due to the lack of training sketch data. To bridge the domain gap we present a novel augmentation technique that is tailored to the task of sketch recognition from a training set of natural images. Randomization is introduced in the parameters of edge detection and edge selection. Natural images are translated to a pseudo-novel domain called “randomized Binary Thin Edges” (rBTE), which is used as a training domain instead of natural images. The ability to scale up is demonstrated by training a CNN-sketch recognition model of more than 2.5 times larger number of categories than used previously. For this purpose, a dataset of natural images from 874 categories is constructed by combining a number of popular computer vision datasets. The categories are selected to be suitable for sketch recognition. To estimate the performance, a subset of 393 categories with sketches is also collected.

I. INTRODUCTION

Free-hand drawings, or sketches, have been a long-lasting means of human communication and expression. Nowadays, the prevalence of digital devices equipped with touch-screens has given free-hand sketches additional roles in a number of educational, business or leisure activities. As a result, computer vision research related to sketches has flourished in a variety of tasks including synthesis [1], [2], perceptual grouping [3], [4], sketch-based image retrieval [5], [6], [7], [8], and sketch recognition [9], [10], [11]. In this paper, we focus on the task of sketch recognition, i.e., how to classify sketches into specific categories. In particular, we target a realistic application scenario, where the number of classes is as large as possible. Prior work commonly keeps the number of classes relatively low. The reason for that is simple - lack of training data. Since annotation effort for sketch recognition includes sketch drawing, this activity becomes prohibitively expensive. For example, the Sketchy dataset [12] required 3,921 hours of sketching for 125 categories. In order to obtain a seven times larger dataset, one would need over 13 human-years of sketch drawing (40 hours a week, 52 weeks a year). Therefore, the task of large-scale sketch recognition requires methods much less demanding on the training data.

To allow scalability in the number of classes, we propose a method that trains a deep network classifier without requiring a single sketch during the training. Instead, only natural images with their labels are used to train the classifier. The method exploits the fact that human-drawn sketches often represent the 2D shape of depicted objects or of their parts. The sketch domain is approximated by detected 2D shapes of objects in natural images. In the following, terms natural images, RGB images, or simply images are used interchangeably.

A novel edge augmentation technique is used to map natural images to a pseudo-novel domain called randomized Binary Thin Edges (rBTE). This augmentation procedure randomly selects an edge detector and an edge selection strategy in order to generate a sketch-like output with different level of details (see Figure 1) and is combined with random geometric augmentations.

Fig. 1. The goal of this work is to recognize sketches at test time (bottom row) without using any sketches at training time. Labeled natural images (top row) are transformed to rBTEs with different level of details (the two middle rows are two instances of rBTE per natural image, thickened for visualization) to bridge the domain gap. Combined with geometric augmentations, the transformed dataset is used to train a deep network for sketch recognition.

Sketch recognition is a standard and well-defined task and, at the same time, collections of annotated images are available. It is possible to cast sketch recognition as a single-source domain generalization with natural images as the source domain. However, we show that a specific approach exploiting the specifics of natural images and sketches brings a relative recognition-rate improvement of more than 20% over unnecessarily generic single-source domain generalization approaches.

The proposed approach is a general augmentation scheme that can include any modern image-to-edge or image-to-sketch.
method. In this work we demonstrate its potential by using
edge detectors [13], [14], [15] trained on an extremely limited
amount of non-sketch data, i.e. 200 natural images. Methods
such as [16], [17] are trained on sketches, and methods such
as [18] require more data. RBTEs form a rich training set
allowing to train, without a single sketch, a CNN-based sketch
classifier, which is the main contribution of this work.

Sketch synthesis is a popular task [19], [20], [21], [22], [23]
in which sketches are generated from images. However, it is
not applicable to the setup that this work explores since these
approaches cannot work without training sketches.

To evaluate the proposed approach we introduce Im4Sketch,
a dataset for large-scale sketch recognition without sketches
for training. It consists of 1,007,878 natural images labeled
into 874 classes used to train the sketch classifier. Testing
is performed on 80,582 sketches coming from 393 classes
that are a subset of the training classes. The dataset is a
composition of existing popular image and sketch datasets,
namely ImageNet [24], DomainNet (DN) [25], Sketchy [12],
PACS [26], and TU-Berlin [27]. The classes are selected,
so that classification by shape is meaningful. For example,
ImageNet categories “Indian elephant” and “African elephant”
are merged into category “Elephant”. The dataset is described
in detail in Section V.

To the best of our knowledge, this is the first work that
delivers sketch recognition of the order of over 800 categories.
The dataset with the second largest number of classes for
sketch recognition is DomainNet [25] with 345 classes, i.e.
more than 2.5 times smaller.

II. RELATED WORK

In this section, we review the prior work on three tasks that
are relevant to the focus of this work, namely sketch recogni-
tion, sketch-based image retrieval, and domain generalization.

a) Sketch recognition: The crowd-sourced free-hand
sketch dataset by Eitz et al. [27] is the first large-scale
dataset on the domain of sketches. Early approaches [28],
[27] focus on adapting hand-crafted features and encodings,
such as SIFT [29], its variants, Fisher vectors [30], and SVM
classifiers to solve the task. The application of deep network
classifiers was stimulated by the enormous effort invested in
annotating sketches. The Sketch-a-Net [31] approach demon-
strates recognition performance surpassing human recognition
ability. This is achieved with a tailored architecture and
a labeled training set of sketches whose size is in the order of
10^6. In the recent work of Qi et al. [32], the combination
of deep and hand-crafted features exhibits very good results.
Some approaches [33], [34] exploit the additional information
of stroke order, when available in the input, to further improve
the recognition accuracy. We assume that this information is
not available and deal with the more general problem.

The scarcity of training data in the domain of sketches is
handled by some approaches by combining sketches and
natural images during the training. Hua et al. [9] attempt
to automatically learn the shared latent structures that exist
between sketch images and natural images. Zhang et al. [35]
transfer the knowledge of a network learned on natural images
to a sketch network. In both these methods the training set
consists of both natural images and sketches. Even though
these approaches are valuable in a few-shot setup, where only
a few labeled sketches per category are available, the setup
with no sketches has not been well studied before with a focus
on the sketch domain. An exception is the recent work of
Lamb et al. [36] where the SketchTransfer task is presented.
Even though their work explores the setup of no available
sketches too, promising results are achieved only when unla-
beled sketches are available during the training. The authors
conclude that the low resolution images of the benchmark is
a limitation. Therefore, in our work we use benchmarks with
higher resolution images that are more realistic.

b) Sketch-based image retrieval: Classical approaches
use edge detection on natural images to bridge the domain gap
and then handle both domains with hand-crafted descriptors
or matching [37], [5], [38], [39], [40]. Deep learning methods
mainly follow a different path. A two branch architecture is
used [41], [42], [43], with a different branch per domain,
where the expectation is to bridge the domain gap based on
large amounts of training data with cross-domain labeling [12].
If learning is involved, the most realistic setup is the zero-shot
sketch-based image retrieval [44]; which is a challenging task,
that is related to, but different from, ours. Radenovic et al. [6]
avoid cross-modal annotation by relying on training labels of
natural images and using edge detection to bridge the domain
gap. Their work focuses on learning shape similarity and does
not attempt to directly generalize to category level recognition.

c) Domain generalization: The most common approach
for domain generalization is invariant feature learning, based
on the theoretical results of Ben-David et al. [45]. Repre-
sentative approaches include kernel-based invariant feature
learning by minimizing domain dissimilarity[46], multi-task
autoencoders that transform the original image to other related
domains, domain classifiers as adversaries to match the source
domain distributions in the feature space [47], [48], and cross-
domain non-contrastive learning as regularization [49].

Some methods specialize for single-source domain gen-
eralization. Examples include hard example generation in
virtual target domains [50], style transfer using auxiliary
datasets [51], and adversarial domain augmentation [52].
Narayanan et al. [53] argue that the shock graph of the contour
map of an image is a complete representation of its shape
content and use a Graph Neural Network as their model.
Wang et al. [54] propose a style-complement module to create
synthetic images from distributions that are complementary to
the source domain.

Data augmentation techniques are commonly used for do-
main generalization. Zhou et al. [55] synthesize data from
pseudo-novel domains under semantic consistency by using a
data generator. Mancini et al. [56] use mixup [57] to combine
different source domains. Carlucci et al. [58] train a model to
solve jigsaw puzzles in a self-supervised manner in addition to
the standard classification loss to improve the generalization
ability of the model.
III. TASK FORMULATION

In this section, we define the task and relate it to existing computer vision tasks. We follow the notation of transfer learning [59] and domain adaptation [60] literature.

A domain \( \mathcal{D} \) is an ordered pair \( \mathcal{D} = (\mathcal{X}, P(X)) \) composed of a space of input examples \( \mathcal{X} \) and a marginal probability distribution \( P(X) \), where \( X \) is a random variable valued in \( \mathcal{X} \). A task \( T = (\mathcal{Y}, P(Y|X)) \) is defined by a label space \( \mathcal{Y} \) and the conditional probability distribution \( P(Y|X) \), where \( Y \) is a random variable with values in \( \mathcal{Y} \).

In the problem, two domains are considered: the target domain \( \mathcal{D}^t = (\mathcal{X}^t, P(X^t)) \) of sketches and the source domain \( \mathcal{D}^s = (\mathcal{X}^s, P(X^s)) \) of natural images, with tasks \( T^t = (\mathcal{Y}^t, P(Y^t|X^t)) \) and \( T^s = (\mathcal{Y}^s, P(Y^s|X^s)) \) respectively. The goal is to learn a predictor \( f : \mathcal{X}^t \rightarrow \mathcal{Y}^t \) for the target domain without having access to any examples from that domain.

The input spaces of both the domains, target and source respectively, are images (RGB, fixed size), thus \( \mathcal{X}^s = \mathcal{X}^t \). The same categories are to be recognized in the two domains, i.e. the label spaces are also identical \( \mathcal{Y}^s = \mathcal{Y}^t \). However, the marginal distributions are significantly different, i.e. \( P(X^s) \neq P(X^t) \).

To make better use of this dataset even during learning predictor \( f \), we make use of the BSDS500 [62] segmentation dataset containing 500 images, 200 of which are for training, consisting of natural images, and the ground truth masks are obtained by manual labeling. This is an additional labeled dataset, not including free-hand sketches of objects, that is indirectly used in our work to improve sketch recognition.

A natural image \( x \) is transformed to \( T(x) \), called rBTE, through a sequence of randomized transformations that are described in the following.

a) Edge detection: is performed to map the input image to an edge-map with values in \([0, 1] \). Various edge detectors are used to extract the edges: The Structures Edges (SE) [13], the Holistically-Nested Edge Detection [14], and the Bi-Directional Cascade Network [15]. While SE uses random forest classifier, the other two are DNN based. Using multiple detectors, the size of available training examples is expanded, in this case by a factor of 3.

All three edge detectors are originally trained with clean edge-maps obtained from ground truth segmentation masks, in particular the BSDS500 [62] segmentation dataset containing only 500 images, 200 of which are for training. The dataset consists of natural images, and the ground truth masks are obtained by manual labeling. This is an additional labeled dataset, not including free-hand sketches of objects, that is indirectly used in our work to improve sketch recognition. Even though not considered in this work, one could possible make better use of this dataset even during learning predictor \( f \) to obtain further improvements.

b) Geometric augmentations: Edge-maps are geometrically transformed with a set of commonly used geometric augmentations for CNNs: Zero-padding to obtain square edge-maps, re-sample to 256\( \times \)256, rotation by angle uniformly sampled in \([-5, 5]\) degrees, crop with relative size uniformly sampled in \([0, 8, 1, 0]\) and aspect ratio in \([\frac{1}{2}, \frac{1}{4}] \) resized to 224\( \times \)224, and finally horizontal flip with a probability of 0.5.

1A reimplementation is used https://github.com/sniklaus/pytorch-hed

IV. METHOD

In this section, we describe the proposed method, the details of the construction of the rBTE domain, as well as the implementation details of the training.

A. Deep network classifier

Predictor \( f : \mathcal{X}^t \rightarrow \mathcal{Y}^t \) is a deep convolutional neural network that takes as input a sketch and predicts the class. It is possible to obtain the predictor by minimizing empirical risk

\[
\frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i), \quad \text{with} \quad (x_i, y_i) \in \mathcal{X}^t \times \mathcal{Y}^t,
\]

when sketches are available during training.

Instead, in this work, the predictor is obtained by minimizing

\[
\frac{1}{n} \sum_{i=1}^{n} \ell(f(T(x_i)), y_i), \quad \text{with} \quad (x_i, y_i) \in \mathcal{X}^s \times \mathcal{Y}^s.
\]

B. Edge Augmentation

A natural image \( x \) is transformed to \( T(x) \), called rBTE, through a sequence of randomized transformations that are described in the following.

a) Edge detection: is performed to map the input image to an edge-map with values in \([0, 1] \). Various edge detectors are used to extract the edges: The Structures Edges (SE) [13], the Holistically-Nested Edge Detection [14], and the Bi-Directional Cascade Network [15]. While SE uses random forest classifier, the other two are DNN based. Using multiple detectors, the size of available training examples is expanded, in this case by a factor of 3.

All three edge detectors are originally trained with clean edge-maps obtained from ground truth segmentation masks, in particular the BSDS500 [62] segmentation dataset containing only 500 images, 200 of which are for training. The dataset consists of natural images, and the ground truth masks are obtained by manual labeling. This is an additional labeled dataset, not including free-hand sketches of objects, that is indirectly used in our work to improve sketch recognition. Even though not considered in this work, one could possible make better use of this dataset even during learning predictor \( f \) to obtain further improvements.

b) Geometric augmentations: Edge-maps are geometrically transformed with a set of commonly used geometric augmentations for CNNs: Zero-padding to obtain square edge-maps, re-sample to 256\( \times \)256, rotation by angle uniformly sampled in \([-5, 5]\) degrees, crop with relative size uniformly sampled in \([0, 8, 1, 0]\) and aspect ratio in \([\frac{1}{2}, \frac{1}{4}] \) resized to 224\( \times \)224, and finally horizontal flip with a probability of 0.5.

1A reimplementation is used https://github.com/sniklaus/pytorch-hed

3597
c) **Thinning - NMS:** Edge thinning is performed by non-maximum suppression that finds the locations with the sharpest change of the edge strength. All values that are not larger than the neighboring values along the positive and negative gradient directions are suppressed. This is a standard process which is also part of the well known Canny detector [63] and commonly used as post-processing for the SE detector.

d) **Hysteresis thresholding:** Hysteresis thresholding, which is another common step of the Canny [63] detector, that transforms the input to a binary images using two thresholds (low and high). Pixels whose value is above (below) the high (low) threshold are (are not) considered edges, while pixels with values between the two thresholds are considered as edges only if they belong to a connected-component containing an edge pixel. Setting a fixed threshold that operates well for the large range of images and objects considered in this work is not possible. Instead, a threshold $t$ is estimated by standard approaches that perform binarization of grayscale images, and set the low and high thresholds to $0.5t$ and $1.5t$, respectively. The thresholding approach is randomly chosen from a candidate pool comprising methods of Otsu [64], Yen [65], Li [66], Isodata [67], and the mean approach [68].

e) **Large connected-components:** In the last stage, pixels belonging to small (less than 10 pixels) connected components, estimated with 8-neighbor connectivity, are discarded.

![Diagram](image.png)

**Fig. 2.** Overview of the training pipeline. Natural images are transformed into rBTEs, which are used with class labels to train a network classifier with cross-entropy loss. The obtained network is used to classify free-hand sketches into the object categories.

**C. Testing**

The relative size of sketches with respect to the image dimensions varies from dataset to dataset. In order to tackle this we perform inference in the following two ways. single-scale testing, which is the testing of each sketch in its original relative size and multi-scale testing. In multi-scale testing the sketch is cropped to its bounding box, padded to have aspect ratio 1:1 and then resized to be 90%, 65% and 45% of the 224x224 network input size. The prediction of the three resized sketches are then ensembled by averaging before the softmax function.

**D. Overview**

A visual overview of the pipeline is shown in Figure 2. The proposed approach uses the sequence of transformations on natural images during the learning stage, and trains a deep convolutional neural network with rBTEs. Randomness is involved in the selection of the edge-map, in the geometric augmentation, and in the selection of the threshold estimation method. Each time an image participates in a batch, only one of the edge-maps and one of the threshold estimation methods is used, chosen with equal probability. A set of thorough ablations is presented in the experimental section by discarding parts of the overall pipeline. During inference, a sketch is fed to the exact same network, after simply performing thinning.

**E. Implementation details**

a) **Training details:** ResNet-101 [69] is used as the backbone network for our ablation study on Sketchy and for the core Im4Sketch experiments. The parameters of the network are initialized with the standard weights from training on ImageNet, i.e. training with rBTEs starts with the network trained on ImageNet with RGB images. Adam optimizer is used with batch size equal to 64 for training on rBTEs. The range test [70] is used to identify the initial learning rate. This process starts with a high learning rate and keeps decreasing it every 50 updates. The initial learning rate is picked in the range of the steepest ascending accuracy on the training set. This method indicates a learning rate of $10^{-3}$ for all of our experiments. The learning rate is decayed by a factor of 10 every 10 epochs for a total of 30 epochs.

b) **Comparison with domain generalization methods:** For a fair comparison with the existing domain generalization methods we train a ResNet-18 [69] with SGD optimizer with momentum 0.9, batch size 64 and learning rate 0.004. We train for 30 epochs without a scheduler.

![Image](image.png)

**Fig. 3.** Image and sketch examples from the Im4Sketch dataset. Class ‘bear’ contains the original ImageNet classes “American Black Bear” and “Ice Bear” whose shape is indistinguishable. Sketches are collected from original datasets with different level of detail as in the case of class “dog”; original datasets are shown at the bottom.
We present a large-scale dataset with shape-oriented set of classes for image-to-sketch generalization called “Im4Sketch”\(^2\). It consists of a collection of natural images from 874 categories for training and validation, and sketches from 393 categories (a subset of natural image categories) for testing.

The images and sketches are collected from existing popular computer vision datasets. The categories are selected having shape similarity in mind, so that object with same shape belong to the same category.

The natural-image part of the dataset is based on the ILSVRC2012 version of ImageNet (IN) [24]. The original ImageNet categories are first merged according to the shape criteria. Object categories for objects whose shape, e.g. how a human would draw the object, is the same are merged. For this step, semantic similarity of categories, obtained through WordNet [71] and category names, is used to obtain candidate categories for merging. Based on visual inspection of these candidates, the decision to merge the original ImageNet classes is made by a human. For instance, "Indian Elephant" and "African Elephant", or "Laptop" and "Notebook" are merged. An extreme case of merging is the new class “dog” that is a union of 121 original ImageNet classes of dog breeds.

In the second step, classes from datasets containing sketches are used. In particular, DomainNet (DN) [25], Sketchy [12], PACS [26], and TU-Berlin [27]. Note that merging is not necessary for classes in these datasets, because the shape criteria are guaranteed since they are designed for sketches. In this step, a correspondence between the merged ImageNet categories and categories of the other datasets is found. As in the merging step, semantic similarity is used to guide the correspondence search. A summary of the common classes per dataset pairs is shown in Table I. Sketch categories that are not present in the merged ImageNet are added to the overall category set, while training natural images of those categories are collected from either DomainNet or Sketchy. In the end, ImageNet is used for 690 classes, DomainNet for 183 classes, and Sketchy for 1 class, respectively. An example of merging and mapping is shown in Figure 3.

Training images of Im4Sketch come from training sets of ImageNet and DomainNet(rel), and from 80% of training set of Sketchy. The validation set is obtained from the validation

\(^2\)Im4Sketch is publicly available on http://cmp.felk.cvut.cz/im4sketch/
We choose the Sketchy benchmark to perform an ablation study for computational reasons. Each ablation includes a 30 epoch training procedure followed by the evaluation; the reported numbers are averaged over five randomized runs. The results of the study are summarized in Table III. Training using single edge-maps, and fixed threshold, without geometric augmentation results in poor performance (ID=1), while simply adding geometric augmentations provides a noticeable boost (ID=2). In these two ablations the threshold is fixed and set equal to the average threshold estimated by the Otsu method on the whole dataset. Then, using all the threshold estimation methods per image increases performance (ID=3). Using all edge-maps (ID=4) provide further boost. This variant constitutes the main approach of this work. However, instead of initializing with the result of training on ImageNet with RGB images, we also perform an experiment where the initialization is the result of training with rBTEs on Im4Sketch (described at the end of this section). This kind of large-scale pre-training is shown very essential (ID=5) and increases performance by 7.5% with single scale and 7.3 in multi scale testing. Two additional ablations show that edge-map binarization is needed (ID=7 versus ID=3), and that without the NMS-based thinning the performance is very low (ID=6). The baseline approach of training on RGB images of the Sketchy dataset and then testing on sketches (ID=0) performs very poorly. This comparison demonstrates the large benefit of rBTEs for the Sketchy dataset which includes detailed and well drawn sketches.

We additionally use the Canny edge detector employed with geometric augmentations, Otsu’s method to adaptively set Canny’s thresholds [73], and Gaussian smoothing with σ randomly chosen in set Canny’s thresholds [73], and Gaussian smoothing with σ randomly chosen in [1, 5]. It achieves 44.9% and 47.9% accuracy in single and multi-scale testing, which is 2.1% and 1.9% lower than HED ablation with no adaptive threshold (ID=2).

### B. Single-source domain generalization comparison on PACS

Table IV summarises the performance of our approach compared to the current state-of-the-art in domain generalization.

| Method   | Photo → Sketch |
|----------|----------------|
| RGB      | 32.6 ± 2.4     |
| SelfReg [49] | 33.7 ± 2.6   |
| L2D [54] | 58.5 ± 3.9     |
| SagNet [72] | 40.7       |

**TABLE IV**

PACS: image-to-sketch generalization.

To the baseline trained directly on RGB natural images. SelfReg [49] performs poorly as expected; it is intended for multi-source domain generalization. For L2D [54] which is designed specifically for the single-source task, we run the provided code and ensure that optimal learning rate according to validation performance on the source task is used; the reported score is reproduced. The reported numbers are averaged over twenty randomized runs. Our approach outperforms all other generic domain generalization methods by a large margin.

### C. Training on Im4Sketch and testing on all

The proposed approach for learning without sketches is to train on the corresponding image training part of Im4Sketch by transforming them into rBTEs. In this way, both the backbone and the classifier are trained only with rBTEs. This is performed either to obtain a sketch classifier for 874 categories, or as pre-training to obtain a better backbone network, tailored for shape-based representation (see ID=5 on Table III).

There are 874 training classes in Im4Sketch, while only 393 classes have sketches for testing, since drawing all the classes is impractical at best. Besides results at single and multi-scale, two evaluating scenarios are reported. First, “argmax over-all”, is the testing over all 874 possible classes. This should be seen as an estimate of the overall performance, as the test-classes are unknown during training. The other scenario, “argmax over-subset” is the testing over the classes that appear in the sketch test set. This corresponds to an unrealistic situation, when posterior probability of classes not in the test set is known to be zero. We only report these results to provide some intuition. The results of our Im4Sketch trained model are summarized in Table V to allow future comparisons. Training on rBTEs performs significantly better than training on RGB images on all setups.

### VII. CONCLUSIONS

In this work, we are the first to train a large-scale sketch classifier that is able to recognize up to 874 categories. Due to the absence of such a large training dataset, the learning is performed without any sketches. Instead, we propose a novel edge augmentation technique to translate natural images to a pseudo-novel domain and use it to train a network classifier. This tailored image-to-sketch method is noticeably better than generic single-source domain generalization approaches.

**Acknowledgement:** This work was supported by MSMT LL1901 ERC-CZ grant, Junior Star GACR grant No. GM 21-28830M, and Grant Agency of the Czech Technical University in Prague grant No. SGS20/171/OHK3/3T/13.
[63] J. Canny, “A computational approach to edge detection,” PAMI, 1986.
[64] N. Otsu, “A threshold selection method from gray-level histograms.” IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, no. 1, pp. 62–66, 1979.
[65] Y. Jui-Cheng, C. Fu-Juay, and C. Shyang, “A new criterion for automatic multilevel thresholding.” IEEE Transactions on Image Processing, vol. 4, no. 3, pp. 370–378, 1995.
[66] C. H. Li and C. Lee, “Minimum cross entropy thresholding,” Pattern Recognition, vol. 26, no. 4, pp. 617 – 625, 1993.
[67] R. TW and C. S, “Picture thresholding using an iterative selection method.” Transactions on Systems, Man, and Cybernetics, vol. 8, no. 8, pp. 630–632, 1978.
[68] C. A. Glasbey, “An analysis of histogram-based thresholding algorithms,” CVGIP, vol. 55, no. 6, pp. 532 – 537, 1993.
[69] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in CVPR, 2016.
[70] L. N. Smith, “Cyclical learning rates for training neural networks,” in WACV, 2017.
[71] C. Fellbaum, WordNet: An Electronic Lexical Database. Bradford Books, 1998.
[72] H. Nam, H. Lee, J. Park, W. Yoon, and D. Yoo, “Reducing domain gap by reducing style bias,” in CVPR, 2021.
[73] M. Fung, G. Yue, and Q. Yu, “The study on an application of otsu method in canny operator,” Proceedings of the 2009 International Symposium on Information Processing, 2009.