TaCo: Textual Attribute Recognition via Contrastive Learning

Chang Nie, Yiqing Hu[†], Yanqiu Qu, Hao Liu, Deqiang Jiang, Bo Ren

Tencent YouTu Lab
{changnie, hooverhu, yanqiuqu, ivanhliu, dqiangjiang, timren}@tencent.com

Abstract

As textual attributes like font are core design elements of document format and page style, automatic attributes recognition favor comprehensive practical applications. Existing approaches already yield satisfactory performance in differentiating disparate attributes, but they still suffer in distinguishing similar attributes with only subtle difference. Moreover, their performance drop severely in real-world scenarios where unexpected and obvious imaging distortions appear. In this paper, we aim to tackle these problems by proposing *TaCo*, a contrastive framework for textual attribute recognition tailored toward the most common document scenes. Specifically, TaCo leverages contrastive learning to dispel the ambiguity trap arising from vague and open-ended attributes. To realize this goal, we design the learning paradigm from three perspectives: 1) generating attribute views, 2) extracting subtle but crucial details, and 3) exploiting valued view pairs for learning, to fully unlock the pre-training potential. Extensive experiments show that TaCo surpasses the supervised counterparts and advances the state-of-the-art remarkably on multiple attribute recognition tasks. Online services of TaCo will be made available.

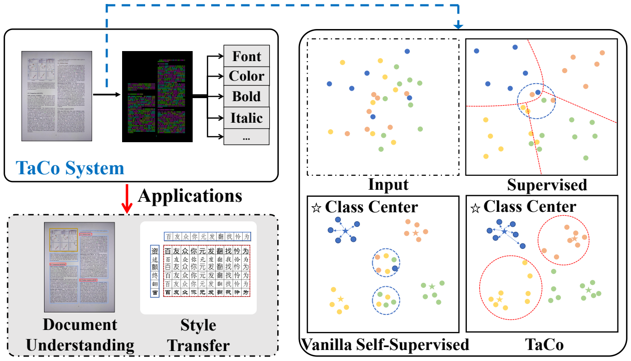

Figure 1: (Left) Precious textual attributes benefit practical applications like document understanding and style transfer. (Right) Semantic spaces obtained from different learning paradigms. Our TaCo system yields aligned attribute representations (red circle) for input with the same attributes beyond supervised approaches and vanilla self-supervised systems, which are constrained by label ambiguity (blue circle).

Introduction

Textual attributes are fundamental in graphic design and also play a key role in forming document styles. For example, in the case of converting a document image into editable formats like Microsoft Word (Wilson and Wilson 2014), retaining the original textual attributes is crucial for user experience. Moreover, graphic designers are keenly interested in identifying attractive styles, like word arts in the wild (Wang et al. 2015). To achieve this goal, they may take photos of the target and turn to experts. However, even for professionals, identifying the correct attributes from a combination of more than 1) 1000+ fonts (Chen et al. 2014), 2) open-ended colors, and 3) other features is error-prone. Hence, an accurate textual attribute recognition (TAR) system is expected to boost versatile applications, as shown in Fig.1.

The design of TAR system is not a trivial task. The reason is mainly twofold: 1) ***Unlimited attributes with subtle details.*** Using the font attribute as an example, it is common to see that the basic difference between pairwise fonts reflected in subtle traits such as letter ending, weight, and slope (Dai et al. 2021), as shown in Fig. 2(a). As fonts are open-ended and ever-increasing through time, the continuously added new types intensified the recognition challenge (Chen et al. 2014). 2) ***Disparite attributes with similar appearance***. What is worse, the real-world input may not be ideal: even scanned PDFs and photographs may contain unexpected distortion that further blur the subtle traits. As the consequence, the missing traits made the different attributes visually similar. Existing methods (Wang et al. 2015; Chen et al. 2021) suffer in these complex scenarios, as shown in Fig. 2(b). To mitigate these gaps, we propose TaCo, the first contrastive framework for textual attributes recognition.

Technical Preview and Contributions. TaCo harnesses contrastive learning with elaborate pretext tasks to fulfill pre-training, allowing the model to learn comprehensive attribute representations over label-free samples. The pretext tasks help to provide better attribute representation, especially for input with subtle changes and noises. To further force the model to focus on local details, we introduce a masked attributes enhancement module (MAEM), which is

[†] Corresponding Author.

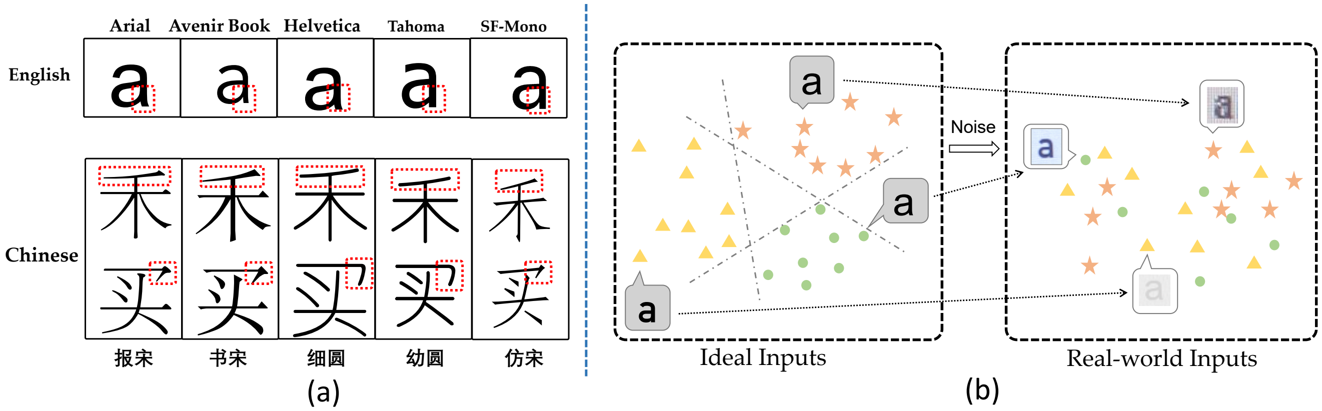

Figure 2: Challenges in textual attribute recognition. (a) Font attributes are rich and the distinction usually lies in local glyphs (red dashed boxes). (b) Existing approaches yield satisfactory result when inputted attributes are distinctive. However, real-world input introduces unexpected noises and makes it difficult for discrimination. Best viewed in color.

achieved by dynamic feature masking operations together with a self-attentive mechanism. MAEM also guides to learn from adjacent characters if existed, which built upon the fact that the attributes of adjacent characters are usually consistent. Finally, we introduce a paired-view scoring module (PSM) to guide the model to learn from high-quality attribute view pairs. Using the generated comprehensive attribute representation as the backbone, we construct a recognition network to yield word-level and even character-level attribute recognition in complex real-world scenarios. The contributions of this paper are summarized as follows:

- We propose a contrastive framework termed TaCo for textual attributes recognition. TaCo is the first textual attribute recognition that supports multiple features at a time: 1) font, 2) color, 3) bold, 4) italic, 5) underline, and 6) strike. The system could be easily extended to support incoming attributes.
- By observing and leveraging the attribute-specific natures, we rigorously design the learning paradigm in respect of 1) generating attribute views, 2) extracting subtle but crucial details, and 3) automatically selecting valued view pairs for learning to ensure the effectiveness of pre-training.
- Experimental results show the superiority of TaCo, which remarkably advances the state-of-the-art of multiple attributes recognition tasks. Online services of TaCo will be publicly released soon to assist relevant researchers and designers.

Related Work

Textual Attribute Recognition

Textual attribute recognition is essentially a fine-grained multi-tagging task, and plays a vital role in many scenarios. Unlike other typical text and entity categorization tasks, attributes rely less on language-specific semantics or layout knowledge, and more on the local details of words (Wang et al. 2015; Huang et al. 2018; Xie et al. 2021). Several traditional methods (Chen et al. 2014; Tao et al. 2015) distinguish attribute classes heavily based on human-customized local feature descriptors and template annotation data, without generality and scalability for commercial applications. Recently, the upsurge of deep learning has dramatically advanced the development of TAR. DeepFont (Wang et al. 2015) firstly exploits CNN for font recognition and obtains favorable results. Moreover, Wang et al. (2018b) introduced transfer learning to address the domain mismatch problem between synthetic and real-world text images, as the prevalent of labeled attributes data scarcity. Chen et al. (2021) designed a local enhancement module that automatically hides the most discriminative features during training to force the network to consider other subtle details. Unfortunately, existing supervised methods remain unsound in real-world scenarios since they failed to tackle label ambiguity and inter-class conflicts brought by image distortion.

Self-Supervised Learning

The self-supervised learning (SSL) allows the model to yield desirable representations from annotation-free data while relieving the burden of labeling (Chen et al. 2020). Consequently, pre-training based on joint multi-modal information has become the common practice for general-purpose document models. For example, the LayoutLM series (Huang et al. 2022) are pre-trained on the IIT-CDIP dataset containing 42 million images, which performs better than routinely training from scratch models. As is well known, the core of SSL involves designing proper pretext tasks and adopting the right evaluation criteria, *e.g.*, masked signal recovery and visual token prediction in Visual-Language model (Huang et al. 2022), and representation consistency of crafted views in contrastive methods (Chen et al. 2020; Chen and He 2021). Nevertheless, empirical evidence suggests (Li et al. 2022) that existing models that learn from the whole documents are prone to capture global structured patterns without desired fine-grained stylistic features, thus inappropriate for attributes recognition tasks.

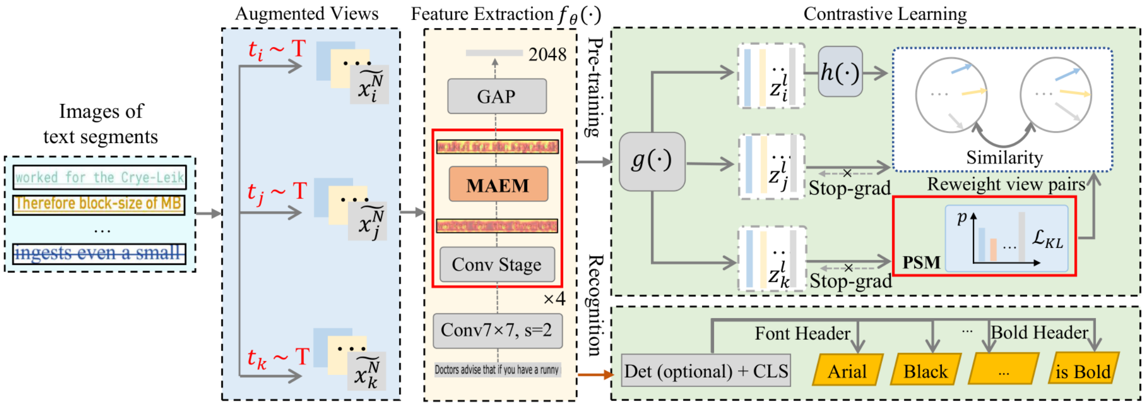

Figure 3: System overview of TaCo. The pre-training pipeline of TaCo is built upon the SimSiam framework and consists of three key designs: 1) generation of augmented attribute views, 2) MAEM-guided feature enhancement, and 3) paired-view scoring module (PSM). $g(\cdot)$ and $h(\cdot)$ are an MLP head and a prediction MLP head, respectively. For the recognition branch, an optional character detection module is introduced to provide character-level attribute recognition. Otherwise, TaCo outputs text-segments-level attributes by default.

Approach

Pre-training

We adopt contrastive learning as the pre-training framework, which learns attribute information implicitly by comparing representations of pairwise positive views. As shown in Fig. 3, the TaCo system is built upon the SimSiam framework (Chen and He 2021) and encapsulates three design refinements: 1) pretext task design, 2) masked attributes enhancement module (MAEM), and 3) paired-view scoring module (PSM). The input of the pre-training system are images of text segments. Text segments are a set of words with random length, which guarantee the sufficient context compared with a single character. With this pre-training paradigm, TaCo is still able to recognizing attribute of a single character accurately, as shown in Section 3.2.

Pretext Task Design. The pretext task design (*a.k.a.* data augmentation) is to construct suitable positive pairs of attributes. For the attribute recognition tasks, they require neither semantic context nor rely on the global visual structure of the inputted images. Hence, popular pre-training tasks including Masked Visual Language Modeling (Xu et al. 2020) and Gaussian blurring (Chen et al. 2020) are not suitable. The former intends to learn from semantic while the latter affects the subtle and crucial feature of attributes. We judiciously design the pretext tasks according to the nature of textual attributes.

Given an input image x, two separate operators t_1, t_2 randomly sampled from the augmentation family $\mathcal{T}$ are applied to x to construct views $\widetilde{x}_i = t_i(x)$, $\widetilde{x}_j = t_j(x)$. For a *minimal sufficient*[1] encoder $f(\cdot)$, the optimal $\mathcal{T}$ is supported to minimize

$$\mathbb{E}_{t_i,t_j\sim\mathcal{T},x}\Big[\underbrace{\Big|\Big|I\big(f(\widetilde{x}_i);f(\widetilde{x}_j)\big)-I(\widetilde{x}_i;\widetilde{x}_j)\Big|\Big|}_{\#1} + \underbrace{d\big(f(x),f(\widetilde{x}_i)\big)+d\big(f(x),f(\widetilde{x}_j)\big)}_{\#2} + \underbrace{I(\widetilde{x}_i;\widetilde{x}_j)-\mathcal{H}(\widetilde{x}_i,\widetilde{x}_j)}_{\#3}\Big] \tag{1}$$

for $\forall\ t_i, t_j \in \mathcal{T}$ and x. Where $d(\cdot)$ denotes certain metrics, *e.g.*, ℓ_1 norm. The first term #1 in the expectation intends to reduce the noisy task-irrelevant mutual-information, and the remaining terms, #2 and #3, maximize the diversity of views with minimal task-relevant information. Hence, we empirically define pretext tasks consisting of three parts: 1) *Random Cropping and Scaling* to take advantage of the content-dependent feature of attributes. The experiments reveal that making the views include varied textual content, or notably task-irrelevant information, is crucial for pre-training. 2) *Color Jittering* is employed to prevent the network from learning trivial task solutions, such as color histogram features. 3) *Random reordering* of characters to prevent the model from learning contextual semantic information. This task is achieved by using synthetic training data introduced in Section 4.1. The synthetic character-level bounding box enables this augmentation.

Masked Attributes Enhancement Module. The Masked Attributes Enhancement Module (MAEM) is designed to achieve better attribute feature fusion in the encoder $f_\theta(\cdot)$. The motivation of MAEM is that adjacent characters in one word share the same attributes with a higher probability. Basing on this observation, MAEM incorporates dynamic masking operations and non-local attention mechanisms (Wang et al. 2018), as shown in Fig. 4. Given a feature tensor $\mathcal{F} \in \mathbb{R}^{H\times W\times C}$, it is partitioned into non-overlapping

[1] An optimal solution of $\arg\min_f I(f(\widetilde{x}_i);\widetilde{x}_i)$ is defined as the minimal sufficient encoder $f^*(\cdot)$ if $I(\widetilde{x}_i;\widetilde{x}_j) = I(f^*(\widetilde{x}_i);\widetilde{x}_j)$ holds (Tian et al. 2020). We use this definition for better problem formation.

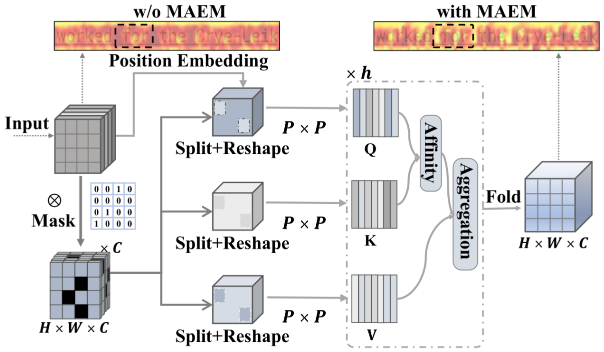

Figure 4: By feature masking and self-attention design, Masked Attributes Enhancement Module (MAEM) guides to learn local details for better attributes feature representation. The attention maps are acquired using Grad-CAM (Selvaraju et al. 2017). Best viewed in color and zoomed in.

patches and randomly masked with probability $p \sim \mathcal{U}(0, \delta)$, where δ is experimentally set as 0.2. This step is performed towards the feature map rather than the input signal, and the masked position varies along the channel dimension. In contrast to *dropout* (Gao, Yao, and Chen 2021), we preserve the spatial continuity of unmasked patches, allowing the network to focus on details rather than global texture information. The mask operation is removed from the inference phase. Next, we utilize multiple convolutional blocks ($1 \times 1\ Conv \rightarrow BN \rightarrow ReLU$) to avoid sharp edges caused by mask operations, and incorporate *patchify* operation[2] ($\texttt{Split+Reshape}$) to obtain the inputs, $\mathcal{K}, \mathcal{Q}, \mathcal{V} \in \mathbb{R}^{h \times \frac{HW}{P^2} \times \frac{CP^2}{h}}$, of the self-attention mechanism. The final output feature tensor $\mathcal{G} \in \mathbb{R}^{H \times W \times C}$ can be computed by:

$$\mathcal{G} = ||_{l=1}^{h} \Big(\hbar_P (\texttt{Softmax}(\frac{\mathcal{Q}_l \mathcal{K}_l^T}{\sqrt{CP^2/h}}) \mathcal{V}_l) \Big), \tag{2}$$

where $||$ denotes concatenation of h attention heads, $\hbar_p$ means recovering feature maps from a sequence of patches of size $P \times P$. We embed the MAEM module behind each Conv stage in the encoder $f_\theta(\cdot)$ to make it more focused on local and contextual information.

Paired-view Scoring Module. We design a Paired-view Scoring Module (PSM) to unleash the learning effectiveness of TaCo, which is parameter-free. The motivation of PSM is that, owing to the randomness of sampling operations and inputs, the generated view pairs are not always guaranteed favorable, and low-quality views will impair the performance. For example, the view produced by random cropping may contain no words or only punctuation with incomplete attributes. In (Peng et al. 2022), authors craft good positive pairs using the heat-map of a network to locate the regions of interest, but this relies on post-processing and handcrafted settings. For an original input image, PSM discriminates the quality of crafted pairs simultaneously during the whole pretraining process.

Formally, given a batch of input images $\{x^l\}_{l=1}^N$, each sample is processed by three randomly sampled data operators $\{t_i^l, t_j^l, t_k^l\} \sim \mathcal{T}$ and the associated augmented views $\widetilde{x}_i^l, \widetilde{x}_j^l$, and $\widetilde{x}_k^l$ are obtained. Note that t_k^l involves only color jittering to maintain view integrity. Then, a parametric function $f_\theta(\cdot)$ (*e.g.*, ResNet-50) and a projection MLP head $g(\cdot)$ transform the views, that is $z_i^l = g(f_\theta(t_i^l))$, and feed their representations $\{z_i^l, z_j^l, z_k^l\}$ into PSM. The computational flow can be presented as:

$$\begin{aligned} &\textbf{Stage } I: \ \widetilde{p_l} = \frac{1}{2}\big(d(z_k^l, z_i^l) + d(z_k^l, h(z_j^l))\big) \\ &\textbf{Stage } II: p_l = -|\widetilde{p_l} - \frac{1}{N}\sum_{l=1}^{N}\widetilde{p_l}| \\ &\qquad \mathcal{P} = \texttt{Softmax}(\{p_l/\tau\}_{l=1}^N), \end{aligned} \tag{3}$$

where τ is a tuning temperature parameter, $h(\cdot)$ is a prediction MLP, and $d(\cdot,\cdot)$ denotes the negative cosine similarity defined as $d(x, y) = -\frac{x}{||x||_2} \cdot \frac{y}{||y||_2}$. On stage I, for each sample, we calculate the similarity of the intact view's feature z_l^k with the two others. Clearly, when encoder $f_\theta(\cdot)$ is sufficient and the cropped views contain adequate or excessive task-relevant information, $\widetilde{p_l}$ up to a scale of 2. On stage II, We zero-meaned $\widetilde{p_l}$ within a batch and take the negative of its absolute value to measure the validity of each pair, where a smaller p_l is better. In this way, the scoring mechanism forces the model to learn from pairs of moderate difficulty rather than those with excessively overlapping or incomplete content views. Then, a $\texttt{softmax}$ function is applied to normalize p_l and output the pairs scores.

In parallel, as shown in Fig. 3, the contrastive branch leverage a prediction MLP $h(\cdot)$ to transform the features of one view, and matches it to another one. The view-invariance of the system is reached by minimizing the negative cosine similarity of the pair representations, and a scored symmetric loss can be formulated as:

$$\mathcal{L}_{cos} = \frac{1}{2}\sum_{l}\mathcal{P}_l\big(d(z_i^l, h(z_j^l)) + d(z_j^l, h(z_i^l))\big), \tag{4}$$

where $\mathcal{P}_l$ is ith element of $\mathcal{P}$ in (3). Note that an important operation $stop\text{-}gradient$ is applied to z_i, z_j before the gradient propagation. We introduce an additional Kullback-Leibler divergence loss to ensure the stability of pre-training, and the final optimization objective is derived as follows

$$\mathcal{L} = \mathcal{L}_{cos} + \lambda\frac{KL(\mathcal{R}||\mathcal{P})}{\log N}, \tag{5}$$

where λ is a trade-off constant, $\mathcal{R}$ denotes the expected uniform distribution of $\mathcal{P}$. The minimum possible value of $\mathcal{L}$ is $\lambda - 1$. The whole framework is trained in an end-to-end manner.

[2] The *patchify* operation was initially designed to exploit the non-local self-correlation properties of infrared images (Gao et al. 2013), and has recently been deployed in vision transformer and MLP architecture designs.

Crop	Color	Shuffle	Pre. (%)	Rec. (%)	F1 (%)
	✓	✓	36.34	33.05	34.62
✓		✓	53.40	56.19	54.76
✓	✓		85.67	90.22	87.89
✓	✓	✓	**87.89**	**90.28**	**89.07**

Table 1: Linear evaluation under composition of data augmentations. "Shuffle" refers to reordering the words, "Crop" means whether the generated views are a local region or full image, and "Color" represents color jittering. "Pre." and "Rec." indicate average precision and recall.

TaCo: Attribute Recognition

The final attribute recognition is built upon the backbone derived from pre-training. Specifically, we train a multi-head linear classifier upon the backbone and apply it to recognize different attributes separately. Now TaCo supports six textual attributes, namely 1) font, 2) color, 3) bold, 4) italic, 5) underline, and 6) strike. More attribute types could be easily extended. The loss function is the sum of the cross entropy between the prediction and ground truth of all attributes, where the weight for font is set as 5 and others as 1, for font attribute learning is more difficult compared with other tasks. Experimental results show that the TaCo system outperforms its counterparts by a large margin.

As the default input of TaCo's pre-training and recognition is text-segments-level, it could be easily extended to character-level with a preposition character detection module. TaCo applies to complex cases with query images that present with totally different textual features. For example, for word "TaCo", character "T" maybe "𝕋" and character "a" maybe "***a***". In this case, character-level attribute detection and classification is needed. We leverage deformable DETR (Zhu et al. 2020) as the detection framework, which outputs the bounding boxes and attributes for each character inside a query image. We take the last three stages outputs of the backbone as the input to the transformer head and finetune the whole network end-to-end until convergence.

Experiments

Datasets. Now there exist no publicly available datasets for textual attributes. We constructed a large-scale synthetic dataset (SynAttr) comprising one million images of text segments for system pre-training and fine-tuning. One-tenth of the data contains words with more than two varying attributes for character-level attribute detection. For each sample, it contains words labeled with a bounding box and six attributes: font, color, italics, bold, underline, and strike. For validation, we manually annotated a dataset Attr-5k comprising 5k individual sentence images, which is cropped from 200 document images with various layouts and page styles collected from real-world scenes. More details of the datasets are given in the supplement.

Implementation. The pre-training of our system is based on the SimSiam framework, with a backbone of vanilla ResNet-50 (He et al. 2016). The standard SGD optimizer with a learning rate of 0.1 is used for optimization. We train for 100 epochs (taking∼26 hours) and adjust the learning rate using a Cosine Annealing strategy. The patch size P and the number of attention heads of MAEM are set to 4. For data augmentation, our pretext tasks include: 1) randomly reordering the words with a probability of 0.5, 2) randomly cropping views from the original image by ratio range (0.8∼1, 0.6∼1, then rescaling and padding them to a fixed size of $(32, 256)$ without changing its aspect ratio, and 3) color jittering alters the brightness, contrast, saturation and hue of an image with an offset degree of (0.4, 0.4, 0.4, 0.1) with a probability of 0.8. In fine-tuning, we remove the data augmentations and retrain the whole network or multiple linear classifiers on frozen features until convergence. Moreover, we build a character detector upon backbone, which training follows the routine setup (Zhu et al. 2020; Li et al. 2022). All experiments are implemented on a platform with 8 Nvidia V100 GPUs.

Methods	Pre. (%)	Rec. (%)	F1 (%)	#Params.
ResNet-50 (vanilla)	85.35	84.21	84.78	23.60 M
+ SE ($r = 16$)	85.93	86.33	86.13	25.42 M
+ CBAM ($r = 16$)	86.62	86.94	86.78	25.63 M
+ MAEM ($\delta = 0$)	86.69	87.35	87.02	23.81 M
+ MAEM ($\delta = 0.2$)	**87.57**	**87.96**	**87.76**	23.81 M
+ MAEM ($\delta = 0.4$)	86.88	86.36	86.62	23.81 M
+ MAEM ($\delta = 0.6$)	85.47	86.46	85.96	23.81 M

Table 2: Ablation study of the plug and play MAEM and comparison with two renowned attention modules. Each model is a single run from scratch.

Methods	Pre. (%)	Rec. (%)	F1 (%)
Random init.	18.57	18.41	18.49
SimSiam (vanilla)	87.89	90.28	89.07
∼ with PSM ($\lambda = 0$)	87.18	90.24	88.68
∼ with PSM ($\lambda = 0.2$)	88.46	91.28	89.85
∼ with PSM ($\lambda = 2$)	**89.52**	**91.34**	**90.42**
∼ with PSM ($\lambda = 10$)	88.11	91.01	89.53

Table 3: Linear evaluation of the scoring module. "Random init." represents random initialization of model parameters instead of loading from pre-training.

Ablation Study and Analysis

In this study, we conduct experiments on Attr-5k to investigate the contribution of individual components in our system. The font attribute is selected for ablation because its recognition is more difficult compared with others. We take the precision, recall, and F1 score as the evaluation criteria.

Pretext Tasks. We analyze the impact of each pretext task on performance. As shown in Table 1, we observe that removing "Crop" augmentation reduces precision, recall, and F1-score of font recognition by 51.55%, 57.23%, and 54.45%, respectively. This concurs with the observation in (Chen et al. 2020). As per predefined guidelines (Section 3.1), we argue that the "Crop" enables pair views to contain different content, thus reducing task-irrelevant mutual-information. For color jittering and words reordering, removal causes a decrease in the precision of 34.63% and 2.22%, respectively, which reflects the importance of view diversity. Overall, incorporating the three tasks yields favor-

	Methods	Font	Color	Italic	Bold	Underline	Strike	**Average**	# Params.	# FLOPs
Baselines	ResNet-50 (He et al. 2016)	86.71	95.78	98.72	90.40	87.10	99.15	92.98	23.60 M	1.34 G
	ResNeXt-101 (Xie et al. 2017)	87.33	94.10	98.01	90.62	86.95	98.63	92.61	42.22 M	2.62 G
	EfficientNet-b4 (Tan and Le 2019)	88.69	**97.43**	98.83	91.00	89.91	98.95	94.12	28.43 M	0.81 G
	Swin-s (Liu et al. 2021)	85.64	97.39	97.51	86.06	84.73	97.62	91.50	48.75 M	8.51 G
	CoAtNet-1 (Dai et al. 2021)	87.39	96.15	97.96	87.84	84.31	98.72	92.06	33.05 M	6.81 G
Variants	DeepFont (Wang et al. 2015)	88.07	96.09	98.28	90.44	86.26	98.99	93.02	23.60 M	1.34 G
	DropRegion (Huang et al. 2018)	88.42	96.29	98.18	91.12	88.68	98.95	93.61	37.88 M	6.45 G
	HENet (Chen et al. 2021)	87.90	95.68	98.48	89.67	88.30	99.03	93.18	23.60 M	1.38 G
Ours	TaCo w/o MAEM	93.25	97.22	99.01	95.18	**90.51**	99.13	95.72	23.60 M	1.34 G
	TaCo	**94.28**	97.06	**99.15**	**96.45**	89.45	**99.35**	**95.96**	23.81 M	1.55 G

Table 4: Comparison with state-of-the-art recognition approaches. "w/o MAEM" is short for "without using the MAEM". We achieved the best recognition performance over other baselines and variants.

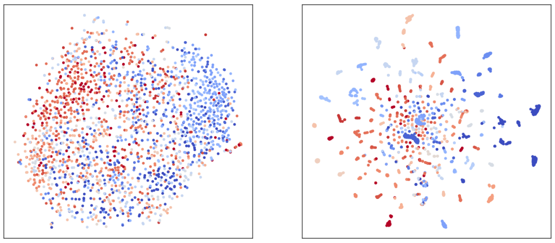

Figure 5: t-SNE visualization of features obtained by supervised learning (left) and TaCo (right). For visualization, we randomly selected 20 fonts from 2k random training images.

able performance and guarantees the pre-training validity. The t-SNE results in Fig. 5 show that the attribute representation shares better aggregation compared with supervised counterpart.

Effect of MAEM. As the MAEM is a component in backbone, it could be used in both supervised learning and SSL (Tables 2 and 4). In the forward stage, the mask ratio of features at random is limited by a hyper-parameter δ. In Table 2, the system equipped with MAEM brings a 2.98 F1-score gain when $\delta = 0.2$, which introduces only 0.21M parameters. Furthermore, even if we set $\delta = 0$, the non-local operation raises the recall by 3.14, showing the necessity to aggregate contextual information. We also compare MAEM with the reputed SE (Hu, Shen, and Sun 2018) and CBAM module (Woo et al. 2018). Empirical results show that MAEM brings significant improvement with little memory overhead. We choose $\delta = 0.2$ in the subsequent experiments.

Effect of PSM. We conduct linear classification to show the benefit of the scoring module, in which the baseline SimSiam is aligned with Table 1. As shown in Table 3, our system obtained varying magnitudes of improvements on the F1-score for different $\lambda > 0$ settings. Notice that when $\lambda = 0$, there is a slight drop in performance, which is probably attributed to the scoring trend shifting toward unsound samples. Notably, for $\lambda = 2$, a precision gain of 1.63% is obtained compared to the vanilla SimSiam, making a better trade-off between the scoring mechanism and the diversity of the learning samples.

Methods	**Pre. (%)**	**Rec. (%)**
Char-Cls (He et al. 2016)	77.5	78.8
∼ with pre-training	89.2	88.6
Deformable DETR (Zhu et al. 2020)	90.8	86.4
∼ EfficientNet-b4 (Tan and Le 2019)	91.4	89.8
∼ RegNet (Radosavovic et al. 2020)	92.1	90.4
TaCo (Ours)	**96.9**	**93.6**

Table 5: Comparison with state-of-the-art methods on character-level font recognition. "Char-Cls" refers to the categorization of each word region individually.

Comparison with the State-of-the-Art

Attributes Recognition. We experimentally demonstrate that the pre-training yields remarkable performance gains for attributes recognition, especially for font. We fine-tune the pre-trained encoder and output the recognition results for six attributes with multiple classifiers. Several strong baselines and other modified variants are choosing for comparison. The evaluation metric is average recognition accuracy. For fairness, all models are trained on the same datasets and settings. Table 4 presents the evaluation results of all methods on Attr-5k. Specifically, the pre-trained ResNet-50 performs far beyond its peers, including EfficientNet (Tan and Le 2019) and variant HENet (Chen et al. 2021), with 5.59% and 5.86% advantages in recognition performance for font. Besides, slight improvements are achieved for other attributes, such as italic and strike. We notice that deeper models (ResNeXt-101) and vision transformers (Swin-S) do not have obvious gain. It is perhaps owing to attribute recognition is focusing more on local details rather than semantic interactions of different regions. As color jittering is crucial for other attributes, the slight inferior performance of color recognition is acceptable for TaCo. Detailed discussion is listed in the supplement.

Overall, our average recognition accuracy over all attributes improves by 2.98 relative to the vanilla ResNet-50 and outperforms its counterparts significantly. This suggests the superiority of our pre-training regime. Albeit our system is learned in terms of fixed categories, the buildup pre-training pipeline allows it to scale to the newly designed classes.

Character-level Attributes Detection. We trained the optional character detector to support character-level attribute recognition. The same training data and settings are reused.

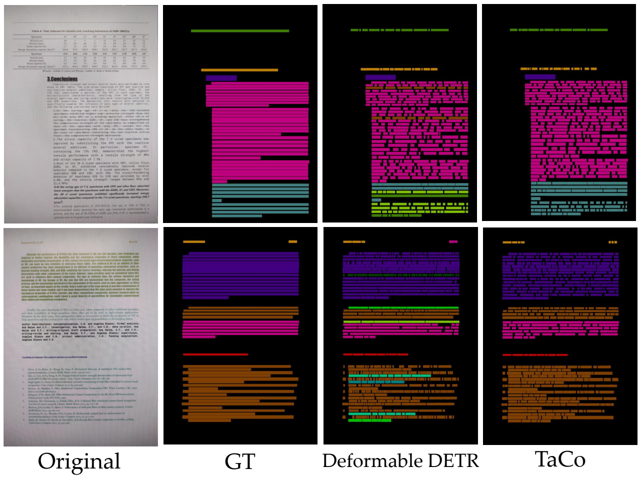

Figure 6: Visualization of character-level font recognition in ordinary document scenarios.

Table 5 shows the average font recognition precision and recall towards various approaches, where detection-based methods are measured with IoU = 0.5. Note that "Char-Cls" means that words are recognized sequentially based on available bounding boxes, and the input is a localized single-word region instead of a whole image. We used deformable DETR as the benchmark and verified the behavior of different backbones separately. We observe that character-wise classification yields poor performance, 1.6% precision lower than the deformable DETR, even with pre-training loaded. This suggests that contextual information benefits the words with indistinct features in the lexicon. For single-stage approaches, our system delivers an accuracy improvement of 5.1% when loading backbone weights, which is more effective than replacing with a stronger baseline. Fig. 6 visualizes the font recognition results for two real-world document images. The TaCo system achieves better accuracy than the supervised counterparts.

Broader Impact of TaCo

Semantic-Entity Labeling in Document. We validate the benefits of the additional provided textual attributes by TaCo on the Form Understanding in Noisy Scanned Documents (FUNSD) dataset (Jaume, Ekenel, and Thiran 2019), which is a well-known challenging task in document understanding. Specifically, we use TaCo to retrieve the attribute information of the text inside an image and embed it into a 512-dim linear space. Then, we construct a 2D attributes grid and sum it with stage-2 output features of ResNet-50. As shown in Fig.7, we can correctly identify entities with the same attributes, such as bolder header and underlined answers, and improve the precision by 1.49. More details are given in the supplement.

Font Generation aims to transfer the style of a reference calligraphy image to ones with different style (Hayashi, Abe, and Uchida 2019), thereby producing characters of a specific font. Existing Method like DG-Font (Xie et al. 2021) already yield satisfactory result, and their performance could be further boosted by treating TaCo as a friend. We observe that these adversarial generative approaches require a powerful discriminator for identifying the font attributes of the synthetic and real samples. Hence, we take the pre-trained encoder of TaCo as the discriminator of DG-Font. As shown in Fig. 8, the generated results of the model with loaded features are more realistic.

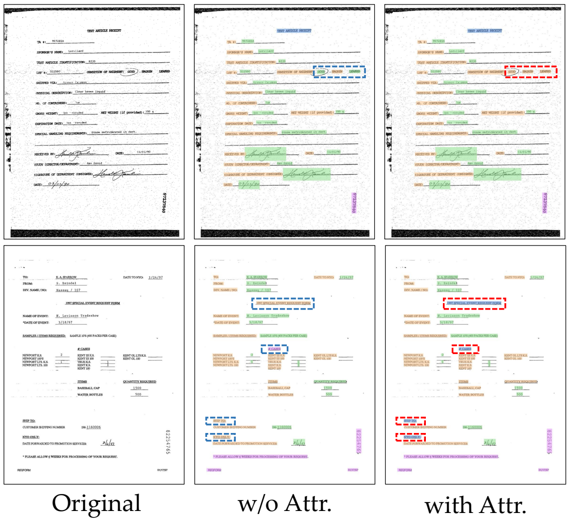

Figure 7: Visualization of document entity recognition. Entities of the same classes are correctly identified with the aid of attribute information (red dashed boxes). "w/o Attr." is short for "without using textual attribute modality".

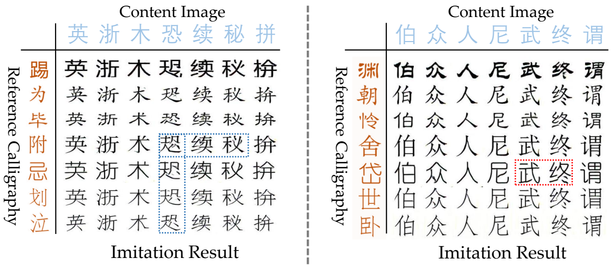

Figure 8: Visualization of font generation. The discriminator of DG-Font with loaded pre-training weights (Right) improves the fidelity of the generated font samples.

Conclusion

This paper presents a novel contrastive framework TaCo for retrieving multiple textual attribute information. By incorporating the attribute-specific characteristics, we rigorously design a pre-training pipeline based on contrastive learning with customized designs to warrant learning effectiveness. Experimental results suggest that our TaCo system is able to learn subtle but crucial features and exhibits superior performance against strong baselines. For future research, we plan to support richer attributes like language classes.

References

Chen, G.; Yang, J.; Jin, H.; Brandt, J.; Shechtman, E.; Agarwala, A.; and Han, T. X. 2014. Large-scale visual font recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3598–3605.

Chen, J.; Mu, S.; Xu, S.; and Ding, Y. 2021. HENet: Forcing a Network to Think More for Font Recognition. In *2021 3rd International Conference on Advanced Information Science and System (AISS 2021)*, 1–5.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Chen, X.; and He, K. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15750–15758.

Dai, Z.; Liu, H.; Le, Q. V.; and Tan, M. 2021. Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34: 3965–3977.

Gao, C.; Meng, D.; Yang, Y.; Wang, Y.; Zhou, X.; and Hauptmann, A. G. 2013. Infrared patch-image model for small target detection in a single image. *IEEE transactions on image processing*, 22(12): 4996–5009.

Gao, T.; Yao, X.; and Chen, D. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Hayashi, H.; Abe, K.; and Uchida, S. 2019. GlyphGAN: Style-consistent font generation based on generative adversarial networks. *Knowledge-Based Systems*, 186: 104927.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141.

Huang, S.; Zhong, Z.; Jin, L.; Zhang, S.; and Wang, H. 2018. DropRegion training of inception font network for high-performance Chinese font recognition. *Pattern Recognition*, 77: 395–411.

Huang, Y.; Lv, T.; Cui, L.; Lu, Y.; and Wei, F. 2022. LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking. *arXiv preprint arXiv:2204.08387*.

Jaume, G.; Ekenel, H. K.; and Thiran, J.-P. 2019. Funsd: A dataset for form understanding in noisy scanned documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, 1–6. IEEE.

Li, J.; Xu, Y.; Lv, T.; Cui, L.; Zhang, C.; and Wei, F. 2022. Dit: Self-supervised pre-training for document image transformer. *arXiv preprint arXiv:2203.02378*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022.

Peng, X.; Wang, K.; Zhu, Z.; Wang, M.; and You, Y. 2022. Crafting better contrastive views for siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16031–16040.

Radosavovic, I.; Kosaraju, R. P.; Girshick, R.; He, K.; and Dollár, P. 2020. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10428–10436.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

Tan, M.; and Le, Q. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, 6105–6114. PMLR.

Tao, D.; Lin, X.; Jin, L.; and Li, X. 2015. Principal component 2-D long short-term memory for font recognition on single Chinese characters. *IEEE transactions on cybernetics*, 46(3): 756–765.

Tian, Y.; Sun, C.; Poole, B.; Krishnan, D.; Schmid, C.; and Isola, P. 2020. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33: 6827–6839.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7794–7803.

Wang, Z.; Yang, J.; Jin, H.; Shechtman, E.; Agarwala, A.; Brandt, J.; and Huang, T. S. 2015. Deepfont: Identify your font from an image. In *Proceedings of the 23rd ACM international conference on Multimedia*, 451–459.

Wilson, K.; and Wilson, K. 2014. Microsoft office 365. *Using Office 365: With Windows 8*, 1–14.

Woo, S.; Park, J.; Lee, J.-Y.; and Kweon, I. S. 2018. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, 3–19.

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1492–1500.

Xie, Y.; Chen, X.; Sun, L.; and Lu, Y. 2021. Dg-font: Deformable generative networks for unsupervised font generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5130–5140.

Xu, Y.; Xu, Y.; Lv, T.; Cui, L.; Wei, F.; Wang, G.; Lu, Y.; Florencio, D.; Zhang, C.; Che, W.; et al. 2020. LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2020. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.