Data Augmentations for Document Images

Yunsung Lee[1]*, **Teakgyu Hong**[2], **Seungryong Kim**[1]
[1]Korea University, [2]Clova AI, NAVER Corp.
swack9751@korea.ac.kr, teakgyu.hong@navercorp.com, seungryong_kim@korea.ac.kr

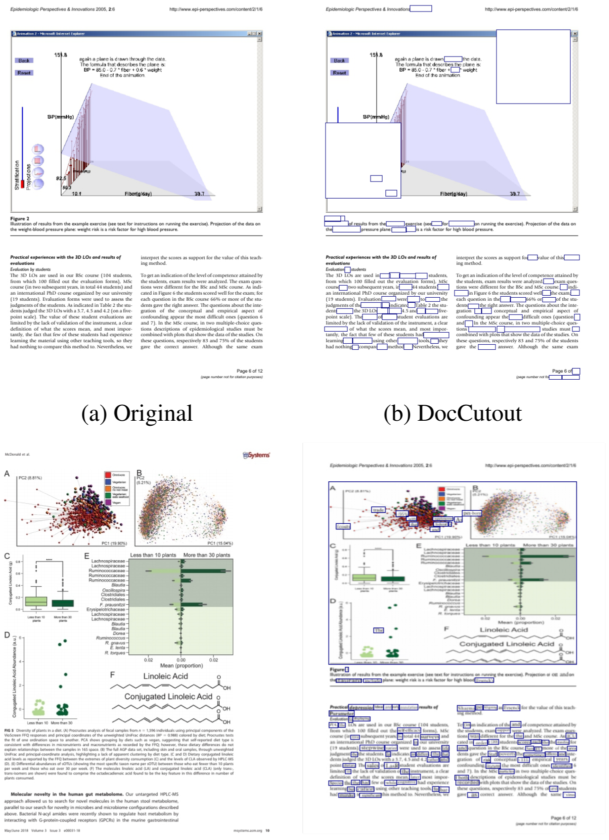
Abstract

Data augmentation has the potential to significantly improve the generalization capability of deep neural networks. Especially in image recognition, recent augmentation techniques such as Mixup, CutOut, CutMix, and RandAugment have shown great performance improvement. These augmentation techniques have also shown effectiveness in semi-supervised learning or self-supervised learning. Despite of these effects and usefulness, these techniques cannot be applied directly to document image analysis, which require text semantic feature preservation. To tackle this problem, we propose novel augmentation methods, DocCutout and DocCutMix, that are more suitable for document images, by applying the transform to each word unit and thus preserving text semantic feature during augmentation. We conduct intensive experiments to find the most effective data augmentation techniques among various approaches for document object detection and show our proposed augmentation methods outperform state-of-the-arts with +1.77 AP in *PubMed* dataset.

1 Introduction

In modern machine learning such as deep neural networks, data augmentation is a de-facto vital solution to augment the limited training data and improve the generalization capability of the models, accounting for the fact that most state-of-the-art models require data at massive scale.

In general, data augmentation greatly contributes to improving the accuracy of the machine learning models, according to the Vicinal Risk Minimization (VRM) principle (Zhang et al. 2017). In computer vision fields, there have been numerous successful approaches, which are formulated by their own strategies. For instance, Mixup (Zhang et al. 2017) used a linear interpolation between two different training instances, and CutMix (Yun et al. 2019) used an image patch cut and paste. (Cubuk et al. 2019; 2020) also presented automated augmentation techniques that is able to search the best combination among several transformations. (DeVries and Taylor 2017) explained that data augmentation with randomly masking part of image, as Cutout, works as a regularizer.



(a) Original

(b) DocCutout

(c) Figure source image

(d) DocCutMix

	Baseline	DocCutout	DocCutMix
AP-figure	91.32	**92.49**	91.88
AP-heading	78.99	**81.90**	81.63
AP-listitem	72.96	**73.99**	73.02
AP-table	92.54	93.52	**93.62**
AP-text	91.87	**92.77**	**92.77**
AP	85.07 (+0.00)	**86.84** **(+1.77)**	86.33 **(+1.26)**

Figure 1: Overview of DocCutout and DocCutMix

*Work done during an internship at Clova AI, NAVER Corp.

Data augmentation has not been used to just improve generalization capability of the model, but can be used to solve the other problems. For instance, (Zhang et al. 2017) showed Mixup can increase the robustness to adversarial examples. (Hendrycks et al. 2019) improved the robustness to perturbation and uncertainty by using both the automated augmented images and the original image by Mixup. For self-supervised learning and semi-supervised learning, data augmentation takes an important role in teaching models about representations from unlabeled data. For instance, consistency regularization (Miyato et al. 2018; Berthelot et al. 2019; Sohn et al. 2020), which trains differently augmented data from the same source data to be classified identically, is the most representative semi-supervised learning methods recently.

Although the data augmentation methods have been popularly proposed and utilized, there were a few attempts to augment a document image. Naturally, in augmenting a document image, the created image must not lose the semantic inforamation of the text area. However, the data augmentation methods described above do not take these points into account.

In this paper, we propose, for the first time, data augmentation methods for the document image analysis. To account for the fact that a document image consists image and text regions having different formats, respectively, we propose a augmentation technique to independently apply transform to each word unit. In particular, we present two methods, DocCutout and DocCutMix, that reinforce the regularization of the model, greatly enhancing the performance.

Our contributions can be summarized as follow:

- We argue that recently studied data augmentations, such as Mixup, Cutout, and CutMix, have a problem in losing word level semantic information in document images.
- We propose two data augmentation methods, namely DocCutout and DocCutMix, to handle word-level images.
- Through various experiments, we show that the proposed DocCutout and DocCutMix are effective and generalize well.

2 Related Work

Data Augmentation

In Computer Vision, data augmentation is a classic and standard way to improve neural networks. Still, various augmentation methods are being published.

Cutout (DeVries and Taylor 2017; Zhong et al. 2020) is an augmentation technique that masks a part of an image. (DeVries and Taylor 2017) explained that Cutout plays a role of regularizer of the model like dropout. Mixup proposed by (Zhang et al. 2017) is a linear interpolation between two data. According to them, Mixup has the effect of Vicinal Risk Minimization. Through this, it is said that not only the accuracy of the model but also the robustness can be obtained.

Mixup is not suitable for localization tasks because features of different classes are mixed throughout the created image. To overcome this, (Yun et al. 2019) proposed CutMix. CutMix replaces some patches of an image with patches of other images, and the target class linearly interpolates with the area ratio of the two images.

Data augmentation is also being studied in the field of NLP (Kobayashi 2018; Wei and Zou 2019; Bari, Mohiuddin, and Joty 2020). (Wei and Zou 2019) improved accuracy in NLP tasks including text classification by using methods such as Synonym Replacement, Random Insertion, Random Swap, and Random Deletion together. BERT (Devlin et al. 2019), a pre-trained language model that has made remarkable developments in various tasks in the NLP field, also performs a kind of data manipluation. BERT performs self-supervised learning by masking some tokens in the input and learning to predict the corresponding parts by the model.

There are a few studies on data augmentation in the field of text image data. In natural scene text image cases, (Gupta, Vedaldi, and Zisserman 2016; Liao et al. 2020; Jaderberg et al. 2014; Wu et al. 2019a) create text images by synthesizing arbitrary text on a natural scene image. These studies will be of great help in improving the performance of text detection, but it is difficult to extend to tasks that require recognition of text semantics in images, such as document layout analysis. If data augmentation is performed for document layout analysis, both visual features of images and the semantic features of text should be kept realistic. Our proposed data augmentation method satisfies this condition through word-based masking or a mix between two data.

Document Layout Analysis

Document layout analysis is a task of identifying the regions of interest in a document image to extract necessary information from the document. There are two approaches to this task, utilizing visual or textual information in the document.

One is to employ the object detection model in the computer vision field. (Hao et al. 2016) and (Schreiber et al. 2017) proposed table detection model in document image based on CNN and Faster R-CNN, respectively. (Soto and Yoo 2019) also utilized Faster R-CNN for object detection, but classified 9 classes including table in document image.

The other is to perform entity extraction in the natural language field. (Katti et al. 2018) encoded document image as a 2D grid of characters and applied fully convolutional encoder-decoder network for information extraction. (Denk and Reisswig 2019; Hwang et al. 2019) proposed a model based on BERT in order to utilize the rich and contextualized word representation of BERT.

The above approaches used only visual or textual information in the document. However, in real documents, such visual and textual information are strongly related in order to represent contents of the documents effectively. Considering the characteristic of a document, it is desirable to perform document layout analysis using both visual and textual information. There are still few studies that consider both information, but because of their desirability, they are being actively studied. (Liu et al. 2019) and (Yu et al. 2020) utilized graph convolution to obtain visual text embeddings and combined them with token embedding to feed combined representation into BiLSTM-CRF model.

3 Methods

In this section, we first introduce previous data augmentation methods tailored to *image* understanding tasks, i.e., Cutout and CutMix, and their limitations when directly applied to *document* image analysis, and then present our data augmentation techniques, called DocCutout and DocCutMix.

Motivation

Cutout Cutout introduced by (DeVries and Taylor 2017) is one of powerful regularization techniques to make deep neural networks generalize better by randomly dropping an input image region, which extends a dropout (Srivastava et al. 2014) working on the input feature itself. It encourages the networks to focus on less discriminative regions on the input, thereby improving such generalization capability.

Specifically, let us denote $\mathbf{X} \in \mathbb{R}^{W\times H\times C}$ and $\mathbf{M} \in \{0,1\}^{W\times H}$ as an input image and a binary mask indicating where to drop out, respectively. In Cutout, new augmented image $\tilde{\mathbf{X}}$ is sampled such that

$$\tilde{\mathbf{X}} = (\mathbf{1} - \mathbf{M}) \odot \mathbf{X} + \mathbf{M} \odot \mathbf{0}, \tag{1}$$

where $\odot$ indicates element-wise multiplication operator. To perform as regularizer, the binary mask $\mathbf{M}$ is randomly sampled with a form of the bounding box coordinates $\mathbf{B} = (r_x, r_y, r_w, r_h)$ such that

$$\begin{aligned} r_x &\sim \text{Unif}\,(0, W)\,, \quad r_w = W\sqrt{\lambda}, \\ r_y &\sim \text{Unif}\,(0, H)\,, \quad r_h = H\sqrt{\lambda}, \end{aligned} \tag{2}$$

where λ denotes the drop ratio. This sampling rule follows the uniform distribution sampling of Unif and thus makes the cropped area ratio $r_w r_h / WH = \lambda$. The binary mask $\mathbf{M}$ is decided by filling with 0 within the bounding box $\mathbf{B}$, 1 otherwise.

CutMix Even though Mixup (Zhang et al. 2017), based on linear interpolation of two different training instances, can greatly improve the model's performance for general classification tasks, it has limited localization ability (Yun et al. 2019), which is the bottleneck to be applied to tasks, e.g., object detection. To overcome this limitation, a new augmentation method, called CutMix, was proposed, where patches within an instance are cut and pasted from another instance and used to train the model with mixed ground-truth labels according to proportion of mixed areas. It has been shown that CutMix takes advantage of both Cutout and Mixup, and outperforms them especially in weakly-supervised object localization task.

Specifically, CutMix generates a new augmented image $\tilde{\mathbf{X}}$ following the rule as

$$\tilde{\mathbf{X}} = (\mathbf{1} - \mathbf{M}) \odot \mathbf{X}_A + \mathbf{M} \odot \mathbf{X}_B, \tag{3}$$

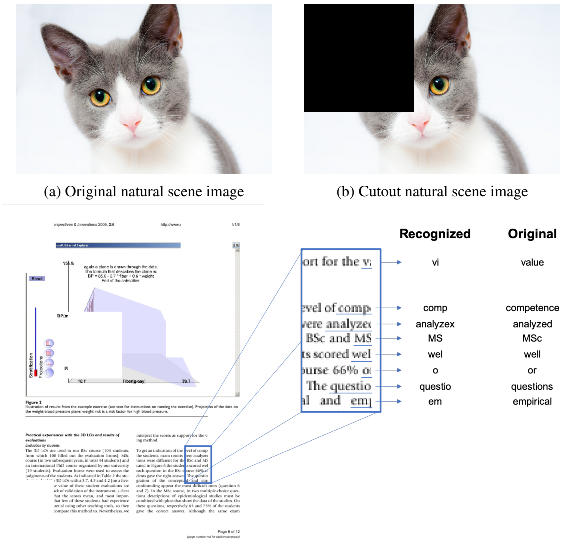
where $\mathbf{X}_A$ is one instance and $\mathbf{X}_B$ is another instance. The new label is determined by taking into account the ratio of mixing as

$$\tilde{y} = (1 - \lambda) y_A + \lambda y_B, \tag{4}$$

where y_A and y_B are ground-truth labels for $\mathbf{X}_A$ and $\mathbf{X}_B$, respectively. The cropping variables are similarly determined as Cuout as follows:

$$\begin{aligned} r_x &\sim \text{Unif}\ (0, W)\ , \quad r_w = W\sqrt{\lambda}, \\ r_y &\sim \text{Unif}\ (0, H)\ , \quad r_h = H\sqrt{\lambda}. \end{aligned} \tag{5}$$

Limitations Although effective to improve the generalization capability of models, aforementioned Cutout and CutMix cannot be directly deployed for tasks requiring *text* units localization, such as document layout analysis, document table detection, and document text detection. In fact, for object detection in an image, a model is generally able to localize an object, even though cropping or occlusion occurs, by focusing on the textures and shapes of remaining parts. However, when localizing and recognizing the text images, the shape of the text is far much more important than the texture, and thus, partially occluded words in the text, by Cutout or CutMix, may not be recovered and recognized completely differently from the original letter. Figure 2 exemplifies this phenomenon. To overcome this limitation, technique separately handling image and text in document image is demanded, which is the topic of this paper.


(a) Original natural scene image

(b) Cutout natural scene image

Recognized	Original
vi	value
comp	competence
analyzex	analyzed
MS	MSc
wel	well
o	or
questio	questions
em	empirical

(c) The OCR results for image with Cutout applied.

Figure 2: Limitations of cutout in document image. (a) the original dog image. (b) An image of a dog whose head is occluded with cutout. However, when looking at the texture and shape of the body, it can still be distinguished as a dog. (c) The OCR results for image with Cutout applied. For the words at the borders of the cropped image area, the OCR results are incorrect. In document images, shape occlusion of text is very critical.

DocCutout

First of all, we present augmentation method, DocCutout, to maintain the text shape in words during augmentation, thus overcoming the limitations of original Cutout. It should be noted that due to the nature of document images, bounding box annotation of word units is relatively easy to obtain. In general, most of the document image datasets were created from latex or xml format metadata (Zhong, Tang, and Yepes 2019; Li et al. 2020b), and thus we naturally access such bounding box annotation. In addition, if not, it is relatively easy to extract the characters and their positions through optical character recognition (OCR) methods (Baek et al. 2019).

DocCutout basically follows the rule of Cutout as in Equation 1, but $\mathbf{0}$ is replaced with the fill_value matrix $\mathbf{F}$ which represents the value to be filled in mask region. We experimented with $\mathbf{F}$ for $\mathbf{0}$, meaning black, and $\mathbf{1}$, meaning white.

$$\tilde{\mathbf{X}} = (\mathbf{1} - \mathbf{M}) \odot \mathbf{X} + \mathbf{M} \odot \mathbf{F}, \tag{6}$$

It is different to Cutout in that we independently cut the region from image and text boxes. Let $\mathbf{B}$ be the bounding box area where masking is attempted, and words($\mathbf{B}$) is the set of word boxes in b. The masking box coordinates are sampled according to:

$$\begin{aligned}
&\text{if label}(\mathbf{B}) == \text{“figure”:}\\
&\quad r_x \sim \text{Unif } (b_x, b_x + b_w)\,, \quad r_w = b_w\sqrt{\lambda},\\
&\quad r_y \sim \text{Unif } (b_x, b_x + b_h)\,, \quad r_h = b_h\sqrt{\lambda},\\
&\text{else:}\\
&\quad i \in \text{sample}(\text{len}(\text{words}(\mathbf{B})), r),\ t^i \in \text{words}(\mathbf{B})\\
&\quad r_x^i = t_x^i, \quad r_w^i = t_w^i,\\
&\quad r_y^i = t_y^i, \quad r_h^i = t_h^i,
\end{aligned} \tag{7}$$

where sample(len(words($\mathbf{B}$)), r) means to sample by the probability ratio of r among the indices of words($\mathbf{B}$).

Such word-by-word masking is not only similar to Cutout, but also similar to the masking method used in BERT (Devlin et al. 2019), which have been proven to effectively train the NLP model in self-supervised fashion. But, it is different in that while word semantic feature vectors are masked in BERT, we mask the visual feature vectors of text images, which allows to learn styles of text such as font and color. Note that some studies attempted to deploy this for document layout analysis through natural language processing of post-OCR data (Xu et al. 2020) or detection modules in the field of Computer Vision (Li et al. 2020a), but they did not utilize visual features or word units at the same time. To achieve further development in document layout analysis, we need to build a unified model that utilizes both visual features and semantic features of text. Unified models such as (Liu et al. 2019) and (Yu et al. 2020) are being studied recently. In this unified model, DocCutOut has a lot of room for application.

DocCutMix

We also present DocCutMix that replaces part of an image with part of another image, inspired by CutMix. The main difference is that it preserves the meaning of word units, by replacing some words or patches in figures in one image from those of other images. Moreover, to preserve the plausibility of the augmented image, the labeled class of the sampled target patch and the original patch should be the same to account for the fact that the styles of the texts vary greatly depending on the class. For instance, the letters of the heading class are usually in bold or colorful, while most general texts are not.

The CutMix can be formulated as follows:

$$\tilde{x} = (\mathbf{1} - \sum_{i=1}^{\|\mathbb{S}\|} \mathbf{M^i}) \odot x + \sum_{i=1}^{\|\mathbb{S}\|} \mathbf{M^i} \odot s\_p(label(s^i)), \tag{8}$$

where $\mathbb{S}$ is the set of original patches sampled with a certain probability from the original image, and s^i is the i-th element of $\mathbb{S}$. This certain probability will be described in detail in Section 4 as a hyper-parameter. The s_p function, short for sample_patch function, returns a selected one patch from all patches in the mini-batch which have the same class as the original patch. Further details of the DocCutMix algorithm are described in Algorithm 1.

4 Experiments

In this section, we report an exhaustive evaluation to assess the effectiveness of proposed augmentation methods, namely DocCutout and DocCutMix, by conducting two main experiments, respectively: 1) ablation study on our augmentation methods and 2) comparison with previous methods.

Experimental Protocol

Dataset To evaluate the proposed methods, we consider a standard benchmark, *PubMed* dataset (Li et al. 2020a). *PubMed* is a subset of PubLayNet (Zhong, Tang, and Yepes 2019), which is one of the large-scale datasets for document object detection, especially sampled from medical journal articles. *PubMed* consists of 12,871 document images and 257,830 bounding boxes with 5 classes such as text, title, list, figure, and table. We train the model on first 9,653 images and evaluate on the remaining 3,218 images. To extract the word bounding boxes, we utilized in-house OCR engine[1] and estimated the label for each word based on the overlap with the area for each class of *PubMed*.

Baseline models for document object detection Following the most recent literature (Li et al. 2020a), we chosen Feature Pyramid Networks (FPN) (Lin et al. 2017a) as a baseline model for document object detection, one of the most effective methods. In particular, FPN exploits the pyramidal feature hierarchy of CNNs and builds a feature pyramid of high-level semantics for all the layers, extracting a mixture of high-level and low-level visual features. It is thus suitable for document image analysis in that the document image often contain both large-scale objects, even taking up most of the image, and small-scale objects, such as a very small listitem with a single word. For FPN, we followed the most common practice and used ResNet-50 as the backbone. We trained the networks with an Momentum SGD optimizer and an initial learning rate of 0.01, which is divided by 10 after 60,000 iterations out of the total 80,000 iterations.

[1] https://clova.ai/ocr

Algorithm 1 Pseudo-code of DocCutMix

```
for each training iteration do
    data_batch = get_minibatch(dataset)
    for each (img, instances) in data_batch do        ▷ img is C×W×H size tensor, instances is a list of (bbox, class, isword)
        for each (bbox, class, isword) in instances do
            if isword or class == 'figure' then        ▷ isword = if bounding box instance indicate word
                patchimage = img[:,bbox[1]:bbox[3],bbox[2]:bbox[0]]
                patchlist[class].append(patchimage)        ▷ There are (# of class) patchlist
    for each img, instances in data_batch do
        for each (bbox, class, isword) in instances do
            if Random(0,1) < mixportion then
                if isword or class == 'figure' then
                    r_i = Unif(0, len(patchlist[class]))
                    ph, pw = (bbox[3]-bbox[1]), (bbox[2]-bbox[0])
                    resized_patch = resize(patchlist[class][r_i],(ph,pw))        ▷ resize patch with interpolation
                    img[:,bbox[1]:bbox[3],bbox[2]:bbox[0]] = resized_patch        ▷ DocCutMix
        instances = [(bbox, class) for (bbox, class, isword) in instances if not isword]        ▷ optional, clear word annotations
```

We further consider two recent methods, the DC5 model proposed in (Dai et al. 2017) and the RetinaNet model proposed in (Lin et al. 2017b). We set the same experimental setting as FPN, namely the same learning rate, optimizer, and ResNet-50 backbone. All models are implemented on top of Detectron2 (Wu et al. 2019b).

Parameters for DocCutout and DocCutMix In all experiments, we set the probability of applying augmentation to 0.5. DocCutOut has hyper-parameters called fill_value, $\sqrt{\lambda}$ and patch_ratio. Since fill_value determines what value to fill the Cutout regions, we considered two cases, namely white (255, 255, 255) and black (0,0,0). $\sqrt{\lambda}$ means the percentage of the cutout part in the figure bounding box, and is determined through $\sqrt{\lambda} \sim \text{Unif}\,(0.3, 0.5)$ for each transform. patch_ratio means the ratio of elements to be Cutout among figures or words in the document, defined for DocCutMix. All data augmentations are implemented on top of Albumentations (Buslaev et al. 2020).

Comparison against baseline augmentations

We conducted experiments on various augmentation methods to determine which method is effective for the document object detection task as follows.

- **Colorjitter:** bright=0.2, contrast=0.2, saturation=0.2, hue=0.2, (standard in Albumentations (Buslaev et al. 2020))
- **Gaussnoise:** (var_limit=(10.0, 50.0), mean=0, (standard setting in Albumentations (Buslaev et al. 2020))
- **Affine:** shift=0.0625, scale=0.01, rotate=2

Colorjitter and Gaussnoise are pixel-level augmentations, so they can be applied directly to document images. In the case of Affine transformation, to preserve the semantic of the text, very small parameters are just used.

Results are given in Table 1. The evaluation metric followed the standard of COCO object detection (Lin et al. 2014). We observe that DocCutout achieves the best result, 86.84 AP. DocCutout outperforms non augmented baseline by +1.77 AP. DocCutMix also showed results that surpassed the methods of other comparison groups. DocCutMix shows the best results in AP-table and AP-text, so it looks good to be applied to table understanding tasks such as document table detection. Moreover, Figure 3 shows that DocCutout and DocCutMix help model converge stable during training. Interestingly, Gaussnoise showed the most unstable convergence graph in training, but it showed superior performance in listitem class which is the most difficult class for all methods. Affine augmentation has also shown competitive results, but there is a threat that affine transformation can transform the semantic of text.

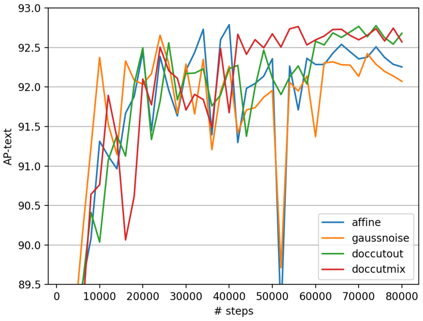


Figure 3: AP-text plot for Affine, Gaussnoise, DocCutout, DocCutMix methods. The convergence graphs of DocCutout and DocCutMix are more stable than those of Affine and Gaussnoise.

Hyper-parameter search for our methods

As previously explained, both DocCutout and DocCutMix have a hyper-parameter called patch_ratio. The experiment was conducted by changing the patch_ratio of the two methods in the order of 0.2, 0.33, and 0.5. As a result, 0.33 for DocCutout and 0.5 for DocCutMix showed the best perfor-

Augmentation	AP-figure	AP-heading	AP-listitem	AP-table	AP-text	AP
Baseline	91.32	78.99	72.96	92.54	91.87	85.07
Colorjitter	91.56	81.31	73.40	92.92	92.42	86.21
Gaussnoise	90.66	80.88	**74.14**	92.79	92.42	86.14
Affine	92.02	81.25	73.06	93.34	92.54	86.32
DocCutout (proposed)	**92.49**	**81.90**	73.99	93.52	**92.77**	**86.84**
DocCutMix (proposed)	91.88	81.63	73.02	**93.62**	**92.77**	86.33

Table 1: Comparison of various augmentations in *PubMed* document object detection
AP is Average Precision at [0.50:0.05:0.95]

Augmentation	patch_ratio	fill_value	AP
DocCutout	0.2	Black	86.44
	0.2	White	86.65
	0.33	White	86.84
	0.5	White	86.61
DocCutMix	0.2		86.11
	0.33		86.27
	0.5		86.33

Table 2: Changing hyper-parameter experiments. DocCutMix doesn't have fill_value parameter. Observing the patch_ratio side, 0.33 for DocCutout and 0.5 for DocCutMix showed the best performance. Since most of the document images in the dataset have a white background, the white fill_value showed better performance.

mance.

DocCutout has another hyper-parameter, fill_value. Since most of the document images in the dataset have a white background, the white fill_value creates more realistic data and showed better AP with +0.21. Table 2 describes the result of hyper-parameter experiments.

Comparison by changing the baseline model

Model	Augmentation	AP
FPN	Baseline	85.07
	DocCutout	**86.84**
FRCNN_DC	Baseline	83.01
	DocCutout	83.60
RetinaNet	Baseline	78.21
	DocCutout	78.30

Table 3: DocCutout's generality to various models

We tested whether DocCutout, which showed the highest AP among the tested augmentation methods, can be applied to various models in general. Table 3 shows the result. The higher the baseline model, the greater the performance improvement when DocCutout was used. The FPN model which have the highest baseline performance showed a performance improvement of +1.77, while the RetinaNet which have the lowest baseline performance showed a performance improvement +0.09.

Combination of data augmentations

Combination of Augmentations	AP
DocCutout	86.65
DocCutout + Affine	86.76
DocCutout + DocCutMix	86.41

Table 4: Combination between DocCutout and other augmentations (patch_ratio $= 0.2$)

Table 4 shows an experiment that combined DocCutMix and Affine augmentation to DocCutout. Affine, which was inferior to DocCutMix in single augmentation, shows better performance in combination with DocCutout. Finding the most appropriate augmentation combination from the data augmentation combination is quite complicated problem. Although it is beyond the scope of our research, it seems possible to find the optimal combination of document image augmentations based on data augmentation methods that we have proposed and experimented with. It is expected that recent studies, such as AutoAugment (Cubuk et al. 2019) and RandAugment (Cubuk et al. 2020), can be applied to solve the problem.

5 Conclusion

Data augmentation plays a variety of roles and contributes greatly to the improved performance of model. However, there have been a lack of the study for data augmentation for document image understanding, which requires understanding both natural language and visual features. In the paper, we have shown that recent data augmentation techniques such as Cutout and CutMix have a limitation and thus cannot be directly applied to document images, although they show a great effectiveness in natural images. To tackle this problem, we proposed two data augmentation methods, DocCutOut and DocCutMix. Our proposed methods show not only performance improvement in *PubMed* dataset, but also generality in various models.

Acknowledgments

We would like to thank Clova AI OCR team, especially Bado Lee, Daehyun Nam and Yoonsik Kim for their helpful feedback and discussion.

References

Baek, Y.; Lee, B.; Han, D.; Yun, S.; and Lee, H. 2019. Character region awareness for text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9365–9374.

Bari, M. S.; Mohiuddin, M. T.; and Joty, S. 2020. Multimix: A robust data augmentation strategy for cross-lingual nlp. *arXiv preprint arXiv:2004.13240*.

Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; and Raffel, C. A. 2019. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, 5049–5059.

Buslaev, A.; Iglovikov, V. I.; Khvedchenya, E.; Parinov, A.; Druzhinin, M.; and Kalinin, A. A. 2020. Albumentations: Fast and flexible image augmentations. *Information* 11(2).

Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, Q. V. 2019. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 113–123.

Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. V. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 702–703.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 764–773.

Denk, T. I., and Reisswig, C. 2019. BERTgrid: Contextualized embedding for 2d document representation and understanding. In *Workshop on Document Intelligence at NeurIPS 2019*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.

DeVries, T., and Taylor, G. W. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Gupta, A.; Vedaldi, A.; and Zisserman, A. 2016. Synthetic data for text localisation in natural images. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Hao, L.; Gao, L.; Yi, X.; and Tang, Z. 2016. A table detection method for pdf documents based on convolutional neural networks. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, 287–292.

Hendrycks, D.; Mu, N.; Cubuk, E. D.; Zoph, B.; Gilmer, J.; and Lakshminarayanan, B. 2019. Augmix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations*.

Hwang, W.; Kim, S.; Seo, M.; Yim, J.; Park, S.; Park, S.; Lee, J.; Lee, B.; and Lee, H. 2019. Post-OCR parsing: building simple and robust parser via bio tagging. In *Workshop on Document Intelligence at NeurIPS 2019*.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Synthetic data and artificial neural networks for natural scene text recognition. In *Workshop on Deep Learning, NIPS*.

Katti, A. R.; Reisswig, C.; Guder, C.; Brarda, S.; Bickel, S.; Höhne, J.; and Faddoul, J. B. 2018. Chargrid: Towards understanding 2d documents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4459–4469.

Kobayashi, S. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 452–457.

Li, K.; Wigington, C.; Tensmeyer, C.; Zhao, H.; Barmpalios, N.; Morariu, V. I.; Manjunatha, V.; Sun, T.; and Fu, Y. 2020a. Cross-domain document object detection: Benchmark suite and method. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 12915–12924.

Li, M.; Xu, Y.; Cui, L.; Huang, S.; Wei, F.; Li, Z.; and Zhou, M. 2020b. Docbank: A benchmark dataset for document layout analysis.

Liao, M.; Song, B.; Long, S.; He, M.; Yao, C.; and Bai, X. 2020. Synthtext3d: synthesizing scene text images from 3d virtual worlds. *Science China Information Sciences* 63(2):120105.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 740–755. Springer.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017a. Feature pyramid networks for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2117–2125.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017b. Focal loss for dense object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2980–2988.

Liu, X.; Gao, F.; Zhang, Q.; and Zhao, H. 2019. Graph convolution for multimodal information extraction from visually rich documents. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Volume 2 (Industry Papers)*, 32–39.

Miyato, T.; Maeda, S.-i.; Koyama, M.; and Ishii, S. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence* 41(8):1979–1993.

Schreiber, S.; Agne, S.; Wolf, I.; Dengel, A.; and Ahmed, S. 2017. Deepdesrt: Deep learning for detection and structure recognition of tables in document images. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, 1162–1167.

Sohn, K.; Berthelot, D.; Li, C.-L.; Zhang, Z.; Carlini, N.; Cubuk, E. D.; Kurakin, A.; Zhang, H.; and Raffel, C. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*.

Soto, C., and Yoo, S. 2019. Visual detection with context for document layout analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3464–3470. Hong Kong, China: Association for Computational Linguistics.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15(1):1929–1958.

Wei, J., and Zou, K. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 6383–6389.

Wu, L.; Zhang, C.; Liu, J.; Han, J.; Liu, J.; Ding, E.; and Bai, X. 2019a. Editing text in the wild. In *Proceedings of the 27th ACM International Conference on Multimedia*, 1500–1508.

Wu, Y.; Kirillov, A.; Massa, F.; Lo, W.-Y.; and Girshick, R. 2019b. Detectron2. https://github.com/facebookresearch/detectron2.

Xu, Y.; Li, M.; Cui, L.; Huang, S.; Wei, F.; and Zhou, M. 2020. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1192–1200.

Yu, W.; Lu, N.; Qi, X.; Gong, P.; and Xiao, R. 2020. PICK: Processing key information extraction from documents using improved graph learning-convolutional networks. In *Proceedings of the 25th International Conference on Pattern Recognition (ICPR)*.

Yun, S.; Han, D.; Chun, S.; Oh, S. J.; Yoo, Y.; and Choe, J. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 6022–6031. IEEE.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; and Yang, Y. 2020. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Zhong, X.; Tang, J.; and Yepes, A. J. 2019. Publaynet: largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 1015–1022. IEEE.