

DocSynth: A Layout Guided Approach for Controllable Document Image Synthesis

Sanket Biswas¹[0000–0001–6648–8270], Pau Riba¹[0000–0002–4710–0864], Josep Lladós¹[0000–0002–4533–4739], and Umapada Pal²[0000–0002–5426–2618]

¹ Computer Vision Center & Computer Science Department
Universitat Autònoma de Barcelona, Spain
{sbiswas, priba, josep}@cvc.uab.es
² CVPR Unit, Indian Statistical Institute, India
umapada@isical.ac.in

Abstract. Despite significant progress on current state-of-the-art image generation models, synthesis of document images containing multiple and complex object layouts is a challenging task. This paper presents a novel approach, called DocSynth, to automatically synthesize document images based on a given layout. In this work, given a spatial layout (bounding boxes with object categories) as a reference by the user, our proposed DocSynth model learns to generate a set of realistic document images consistent with the defined layout. Also, this framework has been adapted to this work as a superior baseline model for creating synthetic document image datasets for augmenting real data during training for document layout analysis tasks. Different sets of learning objectives have been also used to improve the model performance. Quantitatively, we also compare the generated results of our model with real data using standard evaluation metrics. The results highlight that our model can successfully generate realistic and diverse document images with multiple objects. We also present a comprehensive qualitative analysis summary of the different scopes of synthetic image generation tasks. Lastly, to our knowledge this is the first work of its kind.

Keywords: Document Synthesis · Generative Adversarial Networks · Layout Generation.

1 Introduction

The task of automatically understanding a document is one of the most significant and primary objectives in the Document Analysis and Recognition community. Nowadays, especially in business processes, paper scanned and digitally born documents coexist. There is a big variability in real-world documents coming from different domains (forms, invoices, letters, etc.). Modern Robotic Process Automation (RPO) tools in paperless offices have a compelling need for managing automatically the information of document workflows, which can integrate both reading and understanding. According to the standardized recommendations of the Office Document Architecture (ODA) [5], a document representation could be expressed by formalisms that obey two crucial aspects. The

first one considers a document as an image for printing or displaying, while the second one considers its textual and graphical representation for interpreting its layout and logical structure.

The layout structure of a document is fundamentally represented by layout objects (e.g. text or graphic blocks, images, tables, lines, words, characters and so on) while the logical structure describes the semantic relationship between conceptual elements (e.g. company logo, signature, title, body or paragraph region and so on). The recognition of document layout has been one of the most challenging problems for decades. The understanding of layout is a necessary step towards the extraction of information. Business intelligence processes require the extraction of information from document contents at large scale, for subsequent decision-making actions. Many examples can be found in different X-tech areas: fin-tech (analyze sales trends based on intelligent reading of invoices), legal-tech (determine if a clause of a contract has been violated), insurance-tech (liability from accident statement understanding). Document layout syntactically describes the whole document, and therefore allows to give context to the individual components (named entities, graphical symbols, key-value associations). Thus, performance in information extraction is boosted when it is driven by the layout. As in many other domains, the deep learning revolution has open new insights in the layout understanding problem. Consequently, there is a need for annotated data to supervise the learning tasks. Having big amounts of data is not always possible in real scenarios. In addition to the manual effort to annotate layout components, such types of images have privacy restrictions (personal data, corporate information) which prevent companies and organizations to disclose it. Data augmentation strategies are a good solution. Among the different strategies for augmenting data, synthetic generation of realistic images is one of the most successful.

This work discusses a research effort which is intended to develop a synthetic generation tool called DocSynth for rendering realistic printed documents with plausible layout objects desired by the user. A simple illustration of this task is as shown in Figure 1. The proposed model is able to generate samples given a single reference layout image. Thus, it can generate training data with one single sample per class and can be adapted to few-shot settings for document classification tasks. This automatic document image synthesizer could provide a possible solution to manage all papers as well as electronic documents with a centralized platform manipulated by the user. In fact, this practical application has the potential to improve visual search and information retrieval engines. Usually, in retrieval task the user wants to index in a repository of documents (real ones). Instead one could create variations of the query sample to improve retrieval performances of the model.

While classical computer graphics techniques have been used in modeling for example geometry, projections, surface properties and cameras, the more recent computer vision techniques rely on the quality of designed machine learning approaches to learn from real world examples to generate synthetic images. Explicit reconstruction and rendering of document properties (both graphical

and textual) in the form of complex layout objects is a hard task from both computer graphics and vision perspective. To this end, traditional image-based rendering approaches tried to overcome these issues, by using simple heuristics to combine a captured imagery. But applying these heuristical approaches for synthesizing images with complex document layouts generate artifacts and does not provide an optimal solution. Neural rendering brings the promise of addressing the problem of both reconstruction and rendering by using deep generative models like Generative Adversarial Networks (GANs) and Variational Auto encoders (VAEs) and to learn complex mappings from captured images to novel images. They help to combine physical knowledge, e.g., mathematical models of projection and geometry, with learned components to yield new and powerful algorithms for controllable image generation.

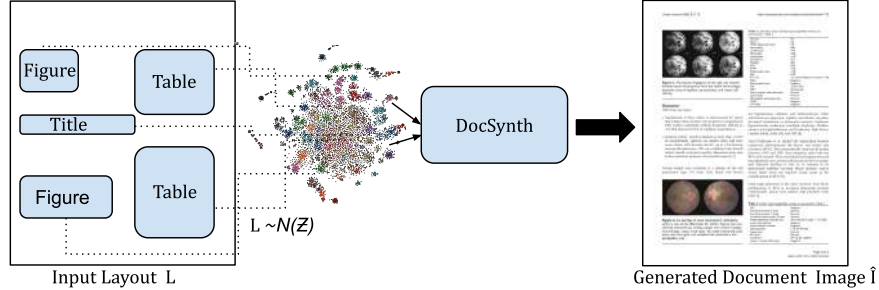


Fig. 1: **Illustration of the Task:** Given an input document layout with object bounding boxes and categories configured in an image lattice, our model samples the semantic and spatial attributes of every layout object from a normal distribution, and generate multiple plausible document images as required by the user.

The main contributions of this work are as follows.

1. A new model is proposed for synthetic document image generation guided by the layout of a reference sample.
2. Qualitative and quantitative results on the PubLayNet dataset [18], demonstrate our model’s capability to generate complex layout documents with respect to spatial and semantic information of object categories.
3. Also this work addresses the layout-guided document image synthesis task with an analytical understanding as the first of its kind in the document analysis community.

The rest of this paper is organized as follows: in Section 2 we review the relevant literature. Section 3 describes the main methodological contribution of

the work. In Section 4 we provide a quantitative and qualitative experimental analysis. Finally, Section 5 draws the main contributions and outlines future perspectives.

2 Related Work

The analysis of structural and spatial relations between complex layouts in documents has been a significant challenge in the field of Document Analysis and Recognition. Extracting the physical and logical layout in documents is a required step in tasks such as Optical Character Recognition for document image transcription, document classification, or information extraction. The reader is referred to [1] for a comprehensive survey on the state of the art on document layout analysis.

As it has been introduced in section 1 the main objective of this work is to construct a generative neural model to construct visually plausible document images given a reference layout. The strategy for augmenting data and its corresponding ground truth by synthetic images automatically generated has gained interest among the Computer Vision community. Since they were proposed by Goodfellow in 2014, Generative Adversarial Networks (GANs) [3] and subsequent variants have been a successful method to generate realistic images, ranging from handwritten digits to faces and natural scenes. A step forward which is a scientific challenge in the controlled generation of images in terms of the composition of objects and their arrangement. Lake et al. in [9] suggested a hierarchical generative model that can build whole objects from individual parts, it is shown to generate Omniglot characters as a composition of the strokes. Zhao et al. [16] proposed a model that can generate a set of realistic images with objects in the desired locations, given a reference spatial distribution of bounding boxes and object labels. Our work has been inspired in this work.

Preserving the reliable representation of layouts has shown to be very useful in various graphical design contexts, which typically involve highly structured and content-rich objects. One such recent intuitive understanding was established by Li et al. [10] in their LayoutGAN, which aims to generate realistic document layouts using Generative Adversarial Networks (GANs) with a wire-frame rendering layer. Zheng et. al. [17] used a GAN-based approach to generate document layouts but their work focused mainly on content aware generation, that primarily uses the content of the document as an additional prior. To use more highly structured object generation, it is very important to focus operate on the low dimensional vectors unlike CNN's. Hence, in the most recent literature, Patil et. al. [13] has come up with a solution called 'READ' that can make use of this highly structured positional information along with content to generate document layouts. Their recursive neural network-based resulting model architecture provided state-of-the-art results for generating synthetic layouts for 2D documents. But their solution could not be applicable for document image level analysis problems. Kang et. al. [6] actually exploited the idea to generate synthetic data at the image level for handwritten word images.

Summarizing, state-of-the-art generative models are still unable to produce plausible yet diverse images for whole page documents. In this work we propose a direction to condition a generative model for whole page document images with synthesized variable layouts.

3 Method

In this section we describe the contributions of the work. We first formally formulate the problem, and introduce the basic notation. Afterwards, we describe the proposed approach, the network structure, its learning objectives, and finally the implementation details.

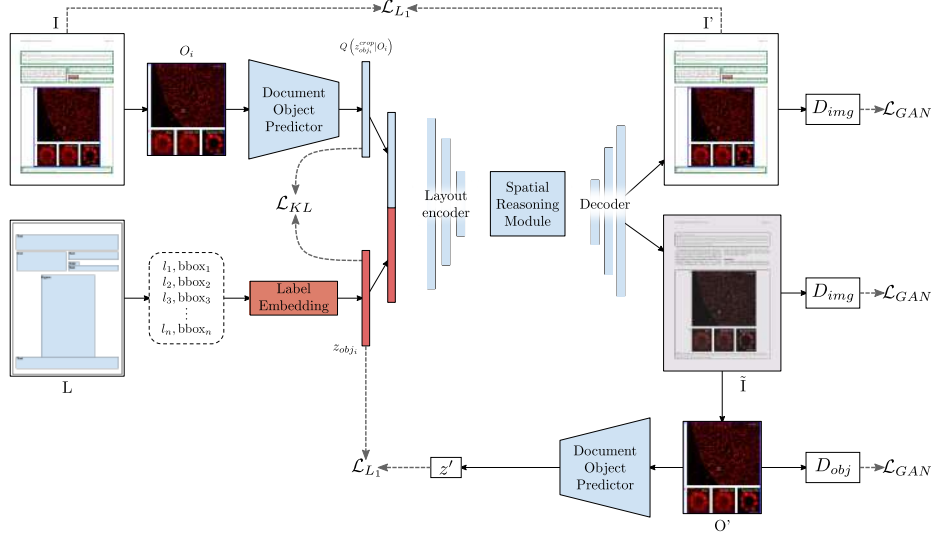


Fig. 2: **Overview of our DocSynth Framework:**The model has been trained adversarially against a pair of discriminators and a set of learning objectives as depicted.

3.1 Problem Formulation

Let us start by defining the problem formally. Let X be an image lattice (e.g. of size 128x128) and I be a document image defined on the lattice. Let $L = \{(\ell_i, \text{bbox}_i)_{i=1}^n\}$ be a layout which contains n labeled object instances with defined class categories $\ell_i \in O$ and bounding boxes of these instances represented by top-left and bottom-right coordinates on the canvas, $\text{bbox}_i \subset X$, and $|O|$ is the total number of document object categories (eg. table, figure, title and so on). Let Z_{obj} be the overall sampled latent estimation comprising every object

instance O_i in the layout L , which can be represented as $Z_{obj} = \{\mathbf{z}_{obj_i}\}_{i=1}^n$. The latent estimation have been sampled randomly for the objects from the standard prior Normal distribution $\mathcal{N}(0, 1)$ under the i.i.d. setup.

The layout-guided document image synthesis task can be codified as learning a generator function G which can map a given document layout input (L, Z_{obj}) to the generated output image \tilde{I} as shown in eqn. 1.

$$\tilde{I} = G(L, Z_{obj}; \Theta_G) \quad (1)$$

where Θ_G represents the parameters of the generation function G which needs to be learned by our model. Primarily, a generator model $G(\cdot)$ is able to capture the underlying conditional data distribution $p(\tilde{I}|L, Z_{obj}; \Theta_G)$ present in a higher dimensional space, equivariant with respect to spatial locations of document layout objects bbox_i .

The proposed model in this work for the above formulated task investigates three different challenges: (1) Given the user provides the input document layout L , is the model capable of synthesizing plausible document images while preserving the object properties conditioned on L ? (2) Given the user provides the input document layout L , can the model generate multiple variable documents using different style \mathbf{z}_{obj_i} of objects while retaining the object configuration ℓ_i , bbox_i in the input layout? (3) Given a tuple (L, Z_{obj}) , is the generator capable of generating consistent document images for different $(\tilde{L}, \tilde{Z}_{obj})$ where a user can add an object to existing layout L or just modify the location or label of existing objects?

Handling such complexities using deep generative networks is difficult due to the difficulty of sampling the posterior elements. This work focuses to tackle the problem by designing a single generator model $G(\cdot)$ that tries to provide an answer to the above mentioned research questions.

3.2 Approach

To build on the layout-guided synthetic document generation pipeline, we aim to explain our proposed approach in two different parts: Training and Inference.

Training: The overall training pipeline of our proposed approach is illustrated in Figure 2. Given an input document image I and its layout $L = \{(\ell_i, \text{bbox}_i)\}_{i=1}^n$, the proposed model creates a category label embedding e_i for every object instance O_i in the document. A set of object latent estimations $Z_{obj} = \{\mathbf{z}_{obj_i}\}_{i=1}^n$ are sampled from the standard prior normal distribution $\mathcal{N}(0, 1)$, while another set of object latent estimations $Z_{obj}^{crop} = \{\mathbf{z}_{obj_i}^{crop}\}_{i=1}^n$ are sampled from the posterior distribution $Q(\mathbf{z}_{obj_i}^{crop} | O_i)$ conditioned on the features received from the cropped objects O_i of input image I as shown in Figure 2 in the document object predictor. This eventually allows us to synthesize two different datasets: (1) A collection of reconstructed images I' from ground-truth image I during

training by mapping the input (L, Z_{obj}^{crop}) through the generator function G . (2) A collection of generated document images \tilde{I} by mapping (L, Z_{obj}) through G , where the generated images match the original layout but exhibits variability in object instances which is sampled from random distribution. To allow consistent mapping between the generated object \tilde{O} in \tilde{I} and sampled Z_{obj} , the latent estimation is regressed by the document object predictor as shown in Figure 2. The training is done with an adversarial approach by including two discriminators to classify the generated results as real or fake at both image-level and object-level.

Inference: During inference time, the proposed model synthesizes plausible document images from the layout L provided by the user as input and the object latent estimation Z_{obj} sampled from the prior $\mathcal{N}(0, 1)$ as illustrated in the Figure 1.

3.3 Generative Network

The proposed synthetic document image generation architecture consists of mainly three major components: two object predictors E and E' , a conditioned image generator H , a global layout encoder C , an image decoder K and an object and image discriminator denoted by D_{obj} and D_{img} respectively.

Object Encoding: Object latent estimations Z_{obj}^{crop} are first sampled from the ground-truth image I with the object predictor E . They help to model variability in object appearances, and also to generate the reconstructed image I' . The object predictor E predicts the mean and variance of the posterior distribution for every cropped object O_i from the input image. To boost the consistency between the generated output image \tilde{I} and its object estimations, the model also has another predictor E' which infers the mean and variances for the generated objects O' cropped from \tilde{I} . The predictors E and E' consist of multiple convolutional layers with two dense fully-connected layers at the end.

Layout Encoding: Once the object latent estimation $\mathbf{z}_i \in \mathbb{R}^n$ has been sampled from the posterior or the prior distribution $\left(\mathbf{z}_i \in \left\{Z_{obj}^{crop}, Z_{obj}\right\}\right)$, the next step is to construct a layout encoding denoted by F_i with the input layout information $L = \{(\ell_i, \text{bbox}_i)_{i=1}^n\}$ as provided by the user for every object O_i in the image I . Each feature map F_i should contain the disentangled spatial and semantic information corresponding to layout L and appearance of the objects O_i interpolated by latent estimation \mathbf{z}_i . The object category label ℓ_i is transformed as a label embedding $e_i \in \mathbb{R}^n$ and then concatenated with the latent vector \mathbf{z}_i . The resultant feature map F_i for every object is then filled with the corresponding bounding box information bbox_i to form a tuple represented by $\langle \ell_i, \mathbf{z}_i, \text{bbox}_i \rangle$. These feature maps encoding this layout information are then fed to a global layout encoder network C containing multiple convolutional layers to get downsampled feature maps.

Spatial Reasoning Module: Since the final goal of the model is to generate plausible synthetic document images with the encoded input layout information, the next step of the conditioned generator H would be to generate a good hidden feature map h to fulfill this objective. The hidden feature map h should be able to perform the following: (1) encode global features that correlate an object representation with its neighbouring ones in the document layout (2) encode local features with spatial information corresponding to every object (3) should invoke spatial reasoning about the plausibility of the generated document with respect to its contained objects.

To meet these objectives, we choose to define the spatial reasoning module with a convolutional Long-Short-Term Memory(conv-LSTM) network backbone. Contrary to vanilla LSTMs, conv-LSTMs replace the hidden state vectors with feature maps instead. The different gates in this network are also encoded by convolutional layers, which also helps to preserve the spatial information of the contents more accurately. The conv-LSTM encodes all the object feature maps F_i in a sequence-to-sequence manner, until the final output of the network gives a hidden layout feature map h .

Image Reconstruction and Generation: Given the hidden layout feature map h already generated by the spatial reasoning module, we move towards our final goal for the task. An image decoder K with a stack of deconvolutional layers is used to decode this feature map h to two different images, I' and \tilde{I} . The image I' is reconstructed from the input image I using latent estimation Z_{obj}^{crop} conditioned on its objects O . The image \tilde{I} is the randomly generated image using Z_{obj} directly sampled from the prior $\mathcal{N}(0, 1)$. Both these images retain the same layout structure as mentioned in the input.

Discriminators: To make the synthetic document images look realistic and its objects noticeable, a pair of discriminators D_{img} and D_{obj} is adopted to classify an input image as either real or fake by maximizing the GAN objective as shown in eqn. 2. But, the generator network H is being trained to minimize \mathcal{L}_{GAN} .

$$\mathcal{L}_{GAN} = \mathbb{E}_{x \sim p_{real}} \log D(x) + \mathbb{E}_{y \sim p_{fake}} \log(1 - D(y)) \quad (2)$$

While the image discriminator D_{img} is applied to input images I , reconstructed images I' and generated sampled images \tilde{I} , the object discriminator D_{obj} is applied at the object-level to assess the quality of generated objects O' and make them more realistic.

3.4 Learning Objectives

The proposed model has been trained end-to-end in an adversarial manner with the generator framework and a pair of discriminators. The generator framework, with all its components help to minimize the different learning objectives during training phase. Our GAN model makes use of two adversarial losses: image

adversarial loss $\mathcal{L}_{\text{GAN}}^{\text{img}}$ and object adversarial loss $\mathcal{L}_{\text{GAN}}^{\text{obj}}$. Four more losses have been added to our model, including KL divergence loss \mathcal{L}_{KL} , image reconstruction loss $\mathcal{L}_1^{\text{img}}$, object reconstruction loss $\mathcal{L}_1^{\text{obj}}$ and auxiliary classification loss $\mathcal{L}_{\text{AC}}^{\text{obj}}$, to enhance our synthetic document generation network.

The overall loss function used in our proposed model can be defined as shown in equation 3:

$$\mathcal{L}_G = \lambda_1 \mathcal{L}_{\text{GAN}}^{\text{img}} + \lambda_2 \mathcal{L}_{\text{GAN}}^{\text{obj}} + \lambda_3 \mathcal{L}_{\text{AC}}^{\text{obj}} + \lambda_4 \mathcal{L}_{\text{KL}} + \lambda_5 \mathcal{L}_1^{\text{img}} + \lambda_6 \mathcal{L}_1^{\text{obj}} \quad (3)$$

3.5 Implementation details

In order to stabilise training for our generative network, we used the Spectral-Normalization GAN [12] as our model backbone. We used conditional batch normalization [2] in the object predictors to better normalize the object feature maps. The model has been adapted for 64x64 and 128x128 image sizes. The values of the six hyperparameters λ_1 to λ_6 are set to 0.01, 1, 8, 1, 1 and 1, respectively. These values have been set experimentally. The Adam optimizer [7] was to train all the models with batch size of 16 and 300,000 iterations in total. For more finer details, we will make our code publicly available.

4 Experimental Validation

Extensive experimentation was conducted to evaluate our adapted DocSynth framework. Since this work introduces the first fundamental approach towards the problem of layout-guided document image synthesis, we try to conduct some ablation studies that are important for proposing our model as a superior baseline for the task. Also, we try to analyse our obtained results with the model, both qualitatively and quantitatively. All the code necessary to reproduce the experiments is available at github.com/biswassanket/synth.doc.generation using the PyTorch framework.

4.1 Datasets

We evaluate our proposed DocSynth framework on the PubLayNet dataset [18] which mainly contains images taken from the PubMed Central library for scientific literature. There are five defined set of document objects present in this dataset: text, title, lists, tables and figures. The entire dataset comprises 335,703 images for training and 11,245 images for validation.

4.2 Evaluation Metrics

Plausible document images generated from layout should fulfill the following conditions: (1) They should be realistic (2) They should be recognizable (3) They should have diversity. In this work, we have adapted two different evaluation metrics for evaluating our rendered images for the problem.

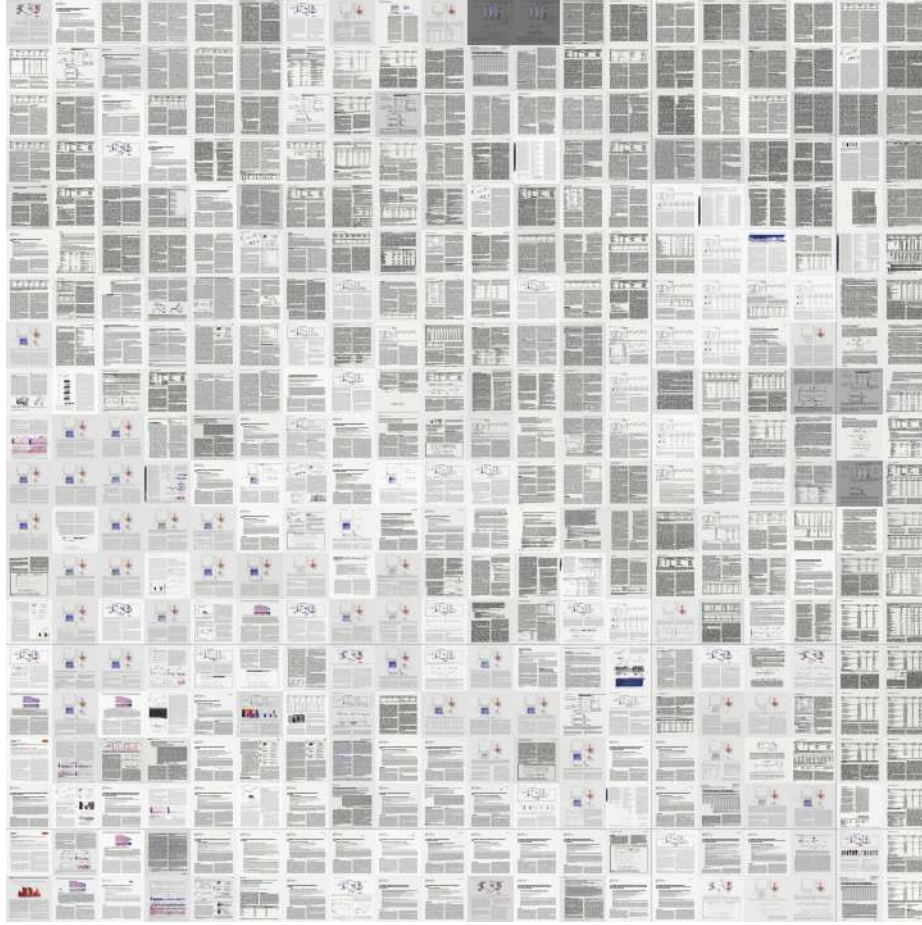


Fig. 3: t-SNE visualization of the generated synthetic document images

Fréchet Inception Distance (FID): The FID metric [4] is a standard GAN performance metric to compute distances between the feature vectors of real images and the feature vectors of synthetically generated ones. A lower FID score denotes a better quality of generated samples and more similar to the real ones. In this work, the Inception-v3 [14] pre-trained model were used to extract the feature vectors of our real and generated samples of document images.

Diversity Score: Diversity score calculates the perceptual similarity between two images in a common feature space. Different from FID, it measures the difference of an image pair generated from the same input. The LPIPS [15] metric actually used the AlexNet [8] framework to calculate this diversity score. In this work, we adapted this perceptual metric for calculating diversity of our synthesized document images.

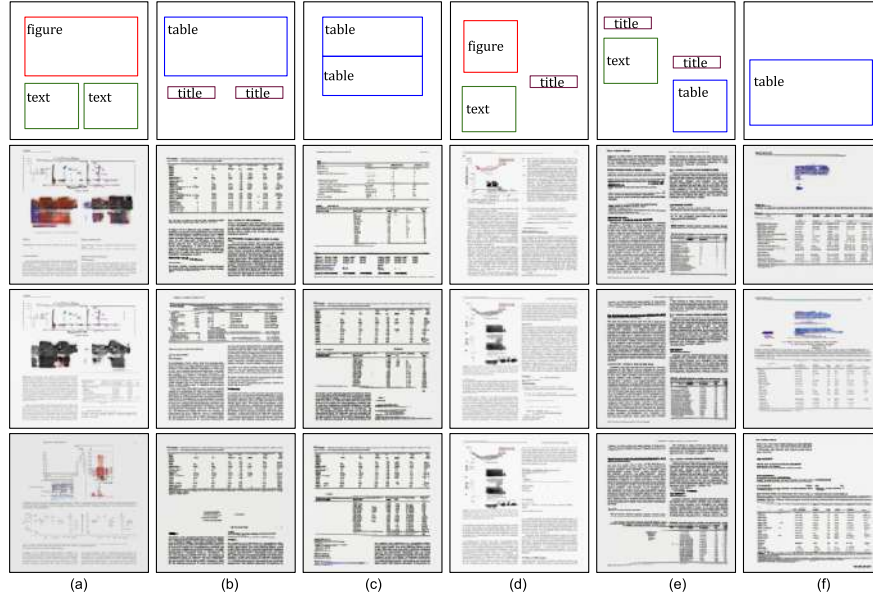


Fig. 4: **Examples of diverse synthesized documents generated from the same layout:** Given an input document layout with object bounding boxes and categories, our model samples 3 images sharing the same layout structure, but different in style and appearance.

4.3 Qualitative Results

Generating Synthetic Document Images: In order to highlight the ability of our proposed model to generate diverse realistic set of document images, we present in Figure 3 a t-SNE [11] visualization of the different synthetic data samples with plausible layout content and variability in overall style and structure. Different clusters of samples correspond to particular layout structure as observed in the figure. We observe that synthetic document samples with complex layout structures have been generated by our model. From these examples, it is also observed that our model is powerful enough to generate complex document samples with multiple objects and multiple instances of the same object category. All the generated samples from the model shown in Figure 3 have a dimension of 128x128. In this work, we propose two final model baselines, for generated images of 64x64 and 128x128 dimension.

Controllable Document Synthesis: One of the most intriguing challenges in our problem study was the controllable synthesis of document images guided by user specified layout provided as an input to our DocSynth generative network. We proposed a qualitative analysis of the challenge in two different case studies.

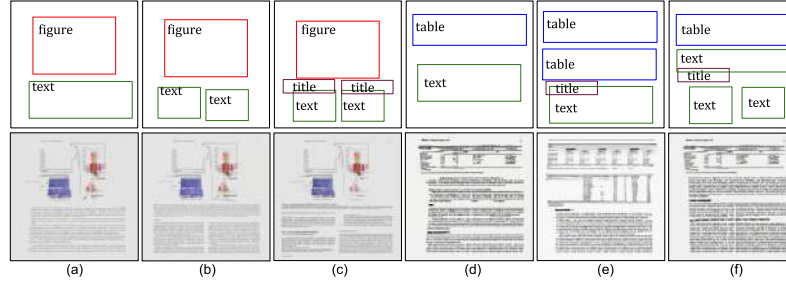


Fig. 5: **Examples of synthesized document images by adding or removing bounding boxes based on previous layout:** There are 2 groups of images (a)-(c) and (d)-(f) in the order of adding or removing objects.

In the first case study as shown in Figure 4, it shows a diverse set of generated documents from a single reference layout as specified by the user. In real life scenario, documents do have the property to exhibit variability in appearance and style while preserving the layout structure. The generated samples in this case obey the spatial constraints of the input bounding boxes, and also the generated objects exhibit consistent behaviour with the input labels.

In the second case study as shown in Figure 5, we demonstrate our model’s ability to generate documents with complex layouts by starting with a very simple layout and then adding a new bounding box or removing an existing bounding box from the input reference layout. From these results we can clearly infer that new objects can be introduced in the images at the desired locations by the user, and existing objects can be modified as new content is added.

4.4 Quantitative Results

Table 1 summarizes the comparison results of the FID and Diversity Score of our proposed model baselines for both 128×128 and 64×64 generated documents. For proper comparison, we have compared the performance scores of the model generated images with the real images. The model generated images are quite realistic as depicted by the performance scores for both FID and Diversity scores. The performance scores obtained for 64×64 image generation model are slightly better compared to those obtained for 128×128 images.

4.5 Ablation studies

We demonstrate the importance of the key components in our model by creating some ablated models trained on the PubLayNet dataset [18]. The following studies clearly illustrate the importance of these elements for solving the task.

Table 1: Summary of the final proposed model baseline for synthetic document generation

Method	FID	Diversity Score
Real Images (128×128)	30.23	0.125
DocSynth (128×128)	33.75	0.197
Real Images (64×64)	25.23	0.115
DocSynth (64×64)	28.35	0.201

Spatial Reasoning module: As already discussed, the spatial reasoning module comprising conv-LSTM to generate the hidden feature map h is one of the most significant components in our model. For generating novel realistic synthetic data, we compare our model results with conv LSTM over vanilla LSTM and also modifying its number of layers for exhaustive analysis.

Table 2: Ablation Study based on different Spatial Reasoning backbones used in our model

Reasoning Backbone	FID
No LSTM	70.61
Vanilla LSTM	75.71
conv-LSTM(k=1)	37.69
conv-LSTM(k=2)	36.42
conv-LSTM(k=3)	33.75

5 Conclusion

In this work, we have presented a novel approach to automatically synthesize document images according to a given layout. The proposed method, is able to understand the complex interactions among the different layout components to generate synthetic document images that fulfill the given layout. Despite the low resolution of the generated images, we believe that this work supposes the first step towards the generation of whole synthetic documents whose contents are related to the context of the page. Indeed, other applications arise from this synthetic generation besides generating realistic images which opens a large variety of future research lines.

The future scope will be mainly focused on two research lines. Firstly, high resolution documents with understandable content is the final goal for any synthetic document generator, therefore, we plan to extend our model towards this end. Secondly, exploiting the generated data for supervision purposes, can im-

prove the performance on tasks such as document classification, table detection or layout analysis.

Acknowledgment

This work has been partially supported by the Spanish projects RTI2018-095645-B-C21, and FCT-19-15244, and the Catalan projects 2017-SGR-1783, the CERCA Program / Generalitat de Catalunya and PhD Scholarship from AGAUR (2021FIB-10010).

References

1. Binmakhashen, G.M., Mahmoud, S.A.: Document layout analysis: a comprehensive survey. *ACM Computing Surveys (CSUR)* **52**(6), 1–36 (2019)
2. De Vries, H., Strub, F., Mary, J., Larochelle, H., Pietquin, O., Courville, A.: Modulating early visual processing by language. *arXiv preprint arXiv:1707.00683* (2017)
3. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Networks. In: *Advances in Neural Information Processing Systems*. pp. 2672–2680 (2014)
4. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. *arXiv preprint arXiv:1706.08500* (2017)
5. Horak, W.: Office document architecture and office document interchange formats: Current status of international standardization. *Computer* **18**(10), 50–60 (1985)
6. Kang, L., Riba, P., Wang, Y., Rusiñol, M., Fornés, A., Villegas, M.: Ganwriting: Content-conditioned generation of styled handwritten word images. In: *Proceedings of the European Conference on Computer Vision*. pp. 273–289 (2020)
7. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems* **25**, 1097–1105 (2012)
9. Lake, B.M., Salakhutdinov, R., Tenenbaum, J.B.: Human-level concept learning through probabilistic program induction. *Science* **350**(6266), 1332–1338 (2015)
10. Li, J., Yang, J., Hertzmann, A., Zhang, J., Xu, T.: Layoutgan: Generating graphic layouts with wireframe discriminators. *arXiv preprint arXiv:1901.06767* (2019)
11. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. *Journal of Machine Learning Research* **9**(11) (2008)
12. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957* (2018)
13. Patil, A.G., Ben-Eliezer, O., Perel, O., Averbuch-Elor, H.: Read: Recursive autoencoders for document layout generation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. pp. 544–545 (2020)
14. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 2818–2826 (2016)
15. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 586–595 (2018)

16. Zhao, B., Yin, W., Meng, L., Sigal, L.: Layout2image: Image generation from layout. *International Journal of Computer Vision* **128**, 2418–2435 (2020)
17. Zheng, X., Qiao, X., Cao, Y., Lau, R.W.: Content-aware generative modeling of graphic design layouts. *ACM Transactions on Graphics* **38**(4), 1–15 (2019)
18. Zhong, X., Tang, J., Yepes, A.J.: Publaynet: largest dataset ever for document layout analysis. In: *Proceedings of the International Conference on Document Analysis and Recognition*. pp. 1015–1022 (2019)