Many-Layer Hotspot Detection by Layer-Attentioned Visual Question Answering

Yen-Shuo Chen[1] and Iris Hui-Ru Jiang[12]
[1]Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan
[2]Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 10617, Taiwan
Email: {spwqee21, huiru.jiang}@gmail.com

***Abstract*—Exploring hotspot patterns and correcting them as early as possible is crucial to guarantee yield and manufacturability. Existing hotspot detection and pattern classification methods consider only the geometry on one single layer or one main layer with adjacent layers. In this paper, we investigate the linkage between many-layer hotspot patterns and potentially induced defect types. We first cast the many-layer critical hotspot pattern extraction task as a visual question answering (VQA) problem: Considering a many-layer layout pattern an image and a defect type a question, we devise a layer-attentioned VQA model to answer whether the pattern is critical to the queried defect type. Furthermore, our layer attention mechanism attempts to identify the relevance of each layer for different defect types. Experimental results demonstrate that the proposed model has superior question-answering ability for modern layouts with more than thirty layout layers.**

I. Introduction

Semiconductor manufacturing defects are generated due to the compounding effects from different process stages in terms of lithography, etching, chemical mechanical polishing, and inter-layer process variations [1]. Hotspots are problematic layout patterns that potentially induce manufacturing defects. These defects can be classified into different types, such as pinching, bridging, necking, line-end pullback, round-off, partial disappearance, etc. Exploring hotspot patterns, identifying their defect types, and correcting them as early as possible is crucial to guarantee yield and manufacturability.

A modern layout contains a number of layers. A hotspot in modern layouts is related to not only the geometry on one specific layer but also the complex inter-layer interactions [2], [3]. State-of-the-art geometry-based hotspot detection and pattern classification methods [4]–[14], however, handle only a single layer or one main layer with adjacent layers. Simply stacking all layers together may dilute the key features of hotspot patterns. Hence, it is not trivial to extend existing works to ascertain the linkage between hotspot patterns and corresponding defect types. In this paper, we attempt to resolve the new many-layer hotspot pattern extraction challenge, which extracts hotspot layout patterns and identifies their potentially induced defect types. Along with the new challenge, the following characteristics of hotspot patterns are observed: (1) The pattern dimension varies greatly for different defect types. (2) Polygons of a hotspot pattern reside over many layers. (3) The importance and relevance of each layer varies for different defect types. (4) Different geometries trigger different manufacturing defects.

We first cast the many-layer hotspot pattern extraction task as a visual question answering (VQA) problem (see Fig. 1). VQA tries to answer a given question based on the visual information, e.g., answering the question "what sport is being played?" for a given sport photo [15]. VQA relies on the good-understanding of both visual and textual content. In the formulated VQA problem, given a many-layer layout pattern (as visual information), and a defect type description (as a question), a model is trained to answer if the pattern is critical to the queried defect type (i.e., possibly triggering this type of manufacturing failure). Based on the characteristics of hotspot patterns and defect type descriptions, we further devise a layer-attentioned VQA model utilizing the importance and relevance of each layer. We train only *one* VQA model to answer *all* defect types. The overall pipeline of our VQA model comprises the following steps: (1) Defect type (question) feature extraction builds up the knowledge of defect descriptions through word embedding [16] and bi-directional long short-term memory (Bi-LSTM). Layout pattern (image) feature extraction extracts pattern features of multi-scale patterns by modified state-of-the-art representations. (2) Layer attention dynamically determines the importance (a weight vector) of layout layers according to the queried defect type feature (serving as a conditional input). (3) Feature fusion joins pattern and defect description features. (4) The classifier predicts if the pattern is critical to the queried defect type based on the fused feature. The main contributions of our work are:

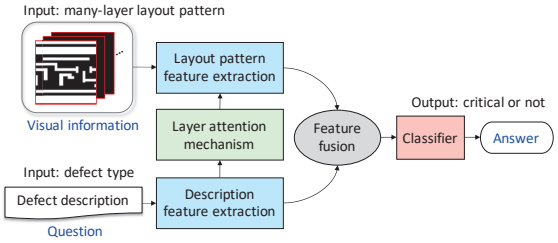

Fig. 1. Overall pipeline of our layer-attentioned visual question answering for hotspot pattern extraction.

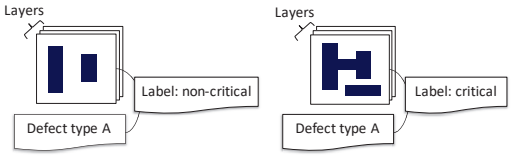

Fig. 2. Two kinds of labels with pattern-defect pairs.

- We address a new many-layer hotspot pattern extraction challenge and ascertain the linkage between hotspot patterns and potentially induced defect types.
- To the best of our knowledge, this is the first work that models the hotspot pattern extraction task as a VQA problem, where an input many-layer layout pattern is the image, and a defect type description is the question.
- We devise a layer-attentioned VQA model to identify the relevance of each layer for different defect types.

Our experiments are conducted on sub-14*nm* modern layouts with more than thirty layout layers. Compared with the conventional VQA model which treats every layer equally, our results show that our VQA model has superior question-answering ability.

II. Problem Statement and Our Flow

A. Problem Statement

We cast many-layer hotspot pattern extraction as a VQA problem:
Problem: Given a many-layer layout pattern (as visual information) and the description of a defect type (as a question), a VQA model is trained to answer whether this pattern is critical to the queried defect type.

Critical Pattern to a defect type is a layout pattern of particular geometry that may trigger a potential manufacturing defect of the queried type. Defect type descriptions in practice are written as a text with special terminology. A non-critical pattern to a defect type can be a non-hotspot pattern or a hotspot pattern which is critical to another type. Due to the variety of defect types, instead of one individual model for each type, our goal is to train *one* VQA model to answer *all* defect types.

B. Evaluation Metrics

In a dataset to train and test a VQA model, as shown in Fig. 2, each data is represented as a triple, containing a pattern-defect pair and a label (answering if the pattern is critical to the specific defect type). The criticality relation between layout patterns and defect types is a many-to-many mapping. A pattern that is critical to one defect type might also be critical to another type and vice versa. Due to the expensive labeling

978-3-9819263-6-1/DATE22/©2022 EDAA

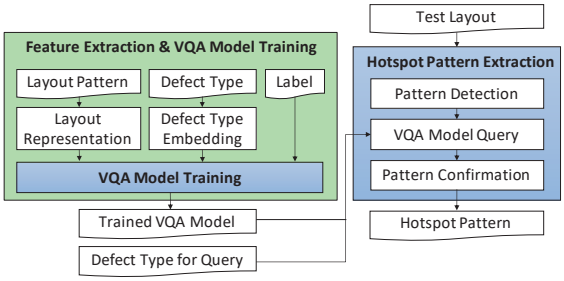

Fig. 3. Overall flow of critical hotspot pattern extraction.

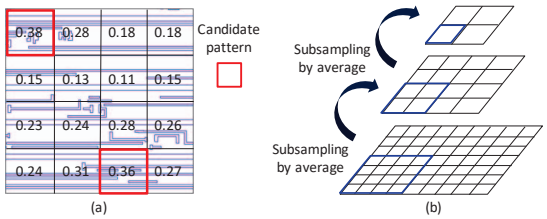

Fig. 4. Pattern detection. (a) Density matrix. (b) Density pyramid.

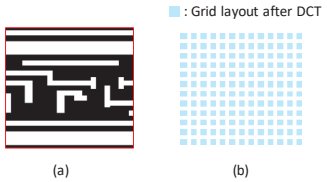

Fig. 5. Single-layer pattern representations: (a) Image. (b) Feature tensor.

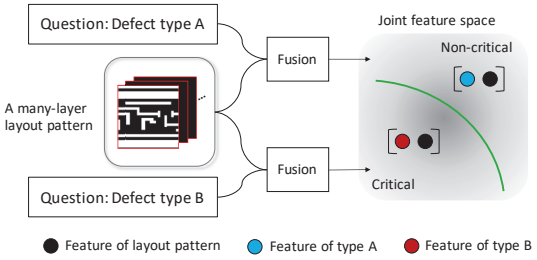

Fig. 6. Feature fusion and classification.

process (process simulation or other failure analysis mechanisms), two disjoint datasets are used in experiments: The training dataset contains partial information of the many-to-many mapping, where each layout pattern is associated with only one defect type with a criticality label. The testing dataset fully labels all pattern-defect pairs for a small set of layout patterns.

We evaluate the prediction performance of the trained VQA model by the following metrics:

Question-Answering Accuracy is the ratio of the number of correctly predicted pattern-defect pairs to the number of all pattern-defect pairs in the testing dataset (with complete mapping).

False Alarm is the number of non-hotspot patterns that are falsely predicted as hotspot patterns.

Miss is the number of hotspot patterns that are falsely predicted as non-hotspot patterns.

A miss is more crucial than a false alarm for extracting hotspot patterns. Hence, we shall target to train a robust VQA model with high question-answering accuracy and low miss for all defect types.

C. Our Hotspot Pattern Extraction Flow

Fig. 3 depicts the overall flow of critical hotspot pattern extraction. First, pattern detection collects candidate patterns of different dimensions from testing layouts based on polygon density distribution. Critical hotspot patterns have variant dimensions for different defect types and are usually located in regions with dense polygons. Thus, multi-scale patterns with higher densities are collected by a density pyramid: As shown in Fig. 4, polygons on all layers are stacked, and a multi-level grid is imposed on the layout. The density matrix and pyramid are constructed by computing the polygon density of each grid on different levels. Second, the trained VQA model answers if a candidate pattern is critical to some defect type. Finally, pattern confirmation checks if the critical pattern is a hotspot pattern.

III. Our VQA Model

This section details our feature extraction and VQA model training.

A. Layout Representation

State-of-the-art layout pattern representations are designed mainly for single-layer patterns. Among them, images and feature tensors as shown in Fig. 5 of all layers are adopted here.

Image: A single-layer layout pattern can be represented as an image. For each layer, the pattern is rescaled to a fixed dimension and quantized into a grayscale image slice.

Feature Tensor: Inspired by image compression, feature tensor extracts frequency domain features while keeping the original spatial information and achieving dimension reduction with little information loss [9]. By discrete cosine transform, each sub-region of a pattern is converted from the spatial domain into the frequency domain. Then, Zig-Zag encoding and thresholding extract low-frequency components which contain most spatial information. Zero padding is performed to match the maximum possible pattern dimension.

B. Defect Type Embedding

A defect type description is written as a text with special terminology. The simplest way is one-hot encoding; however, because the encoding of each word is independent, one-hot encoding has no contextual and semantic information embedded. Distributed representations, e.g., Word2vec and Doc2vec, cluster similar words or sentences in a much lower vector space than one-hot encoding and thus can learn the context of every word/sentence in a document, semantic and syntactic similarity, and relation with other words/sentences.

Word2vec: Word2vec [16] adopts unsupervised learning and produces a low dimensional vector space. Each unique word in the input corpus is added to vocabulary and assigned to a vector in the space. Word2vec has two network architecture options to produce a distributed representation of words: (1) Continuous bag-of-words (CBOW) predicts the current word from a window of surrounding context words. (2) Continuous skip-gram uses the current word to predict the surrounding context words of the window. The skip-gram architecture attach more weight to nearby context words than to distant context words. Because the descriptions of defect types are quite different from the words used in daily life, a pre-trained model cannot be applied here. CBOW makes prediction according to the word frequency, while continuous skip-gram predicts frequent or infrequent word equally. Hence, skip-gram can learn rare words better. Because many infrequent words are used in defect types, Word2vec with continuous skip-gram is more suitable for defect types.

Doc2vec: Doc2vec [17] is an unsupervised model towards learning sentence embedding, and it can learn fixed dimensional text features from variable-length sentences. By adding a document-ID, Doc2vec is an extension of Word2vec that encodes entire sentences as opposed to individual words.

C. Feature Fusion

Learning effective fusion of visual and textual features is the key to visual question answering, e.g., concatenation, element-wise multiplication, etc. In our task, as shown in Fig. 6, if a layout pattern is non-critical to defect type A but critical to defect type B, the pattern and types A, B are fused after feature extraction and transformed into joint embedding space; then, the classifier answers critical for the pattern queried with type B, and non-critical for the pattern queried with type A.

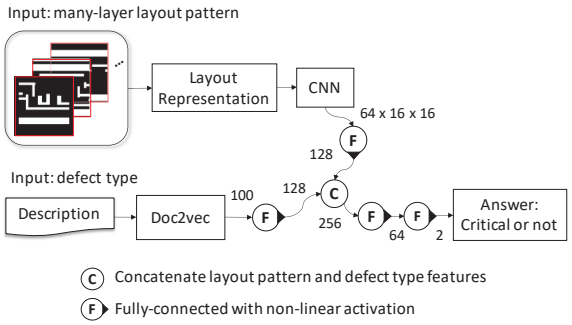

Fig. 7. The baseline VQA model.

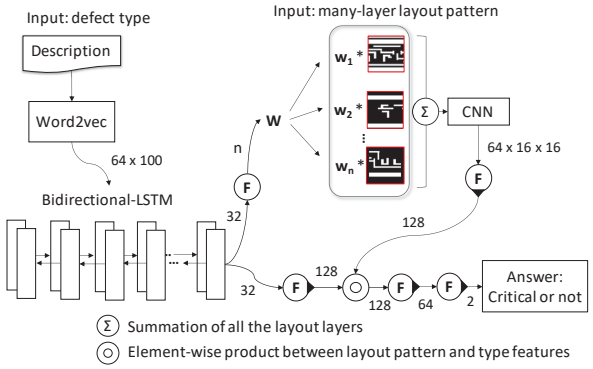

Fig. 8. Our layer-attentioned VQA model.

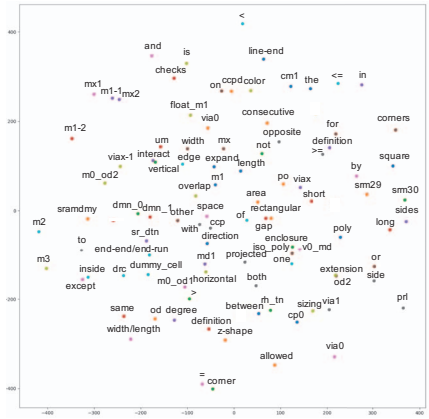

Fig. 9. Visualizing word embedding of defect types using t-SNE.

D. VQA Model

In this subsection, we introduce a commonly used VQA model, which treats every layer equally, and propose our layer-attentioned VQA model, which pays attention to different layers for different types.

1) The Baseline VQA Model: VQA is simplified to a classification problem in most VQA models, e.g., [15], [18]. The number of answers in the training set is the number of final classes that the model learns to predict. The general architecture contains two branches: Given a visual feature as the input to the visual network branch, and given a question feature as the input to the question network branch, the outputs of two branches fuse together before being fed into the classifier to predict the answer class. Hence, this VQA model is essentially a multi-class classification model. Based on the general architecture, Fig. 7 shows a baseline VQA model for our task. The input defect type is transformed into a 100-dimensional defect type embedding by trained Doc2vec as sentence embedding layer, and then fed to a fully connected layer for producing a 128-dimensional defect type feature. Meanwhile, the layout representations of all layers are stacked together and fed to a convolution neural network (CNN) with a fully connected layer to extract a 128-dimensional layout feature. Finally, the defect type feature and layout feature are concatenated and fed to a softmax layer to predict whether the layout pattern is critical to the queried defect type.

2) Our Layer-Attentioned VQA Model: The baseline VQA model incurs two issues: One observed issue is that Doc2Vec has a relatively weaker prediction capability for a long defect type description which is not included in the training set. Therefore, we combine Word2vec with a recurrent neural network (RNN), bidirectional long short-term memory (Bi-LSTM) [19], [20], to extract the information of each word in a sentence sequentially, and then embed the sentence into a semantic vector. By doing so, our model is extendable to handle other defect types with different or complex descriptions using the words we have learned.

The other issue is that the baseline VQA model treats every layer in a many-layer layout pattern equally. For identifying critical hotspot patterns, the importance of each layer should be different for different defect types. Hence, treating every layer equally will dilute key information for learning and results in a negative impact on model prediction. To resolve this issue, we devise a *layer attention* mechanism to focus on the layers which are important to the queried defect type. As depicted in Fig. 8, the defect type feature vector (generated by Word2vec with Bi-LSTM) is fed into a fully connected layer with linear activation to produce a layered relevance weight vector w of dimension n, where n denotes the total number of layers in a layout pattern (image or feature tensor representation *without* stacking all layout layers). The learned layered relevance weights capture the essence of defect type descriptions thus improving question answering.

Before training our VQA model, we train a Word2vec model with all defect type descriptions for embedding each word. We represent a word with a 100-dimensional word vector. Part of word vectors of defect types of a sub-14*nm* process are visualized by t-SNE [21] as shown in Fig. 9, where words with similar context occupy close spatial positions in the latent space. Next, for embedding the defect type description, we limit the text length of a defect type to 64 words by dummy filling or thresholding. Hence, a defect type as the question can be transformed into a word embedding with 64×100 dimensions, and then it is transformed into a 32-dimensional defect type embedding by Bi-LSTM. The defect type embedding is fed into the visual network for layer attention and to the question network for feature extraction simultaneously.

In the question network, the 32-dimensional defect type embedding is fed into a fully connected layer with a 128-dimensional output for defect type feature extraction. In the visual network, the embedding of a defect type is fed to a fully connected layer with linear activation to produce n layered relevance weights. Next, each layer of the input layout pattern after layout representation is multiplied by the corresponding layered relevance weight, and then all layers are stacked into a layer and fed to the CNN network with a fully connected layer to extract a 128-dimensional layout feature. Finally, the layout pattern feature and the defect type feature are fused by element-wise product for prediction.

E. Transfer Learning for a New Technology Node

Transfer learning [22] transfers the knowledge from a source domain to a target domain. We can start with a source model (our layer-attentioned VQA model) and generate the target model with a limited amount of data from a new technology node if the layout geometry has no fundamental changes. First, we retrain the Word2Vec model with the new defect type descriptions to obtain the new word embeddings. Next, we use the new word embeddings and patterns to fine-tune the trained VQA model.

IV. Experimental Results

The dataset was collected from 7 layouts with sub-14*nm* process. We apportion the dataset into training and testing sets with an 80-20 split. 57 defect types were selected as queried types. There are 38 layout layers that are possibly involved in the selected defect types. A total of $79,593$ triple data were included. In the testing set, we labeled the 480 patterns to 57 defect types; $27,360$ pattern-defect pairs are generated. We implemented the network architectures of baseline VQA model

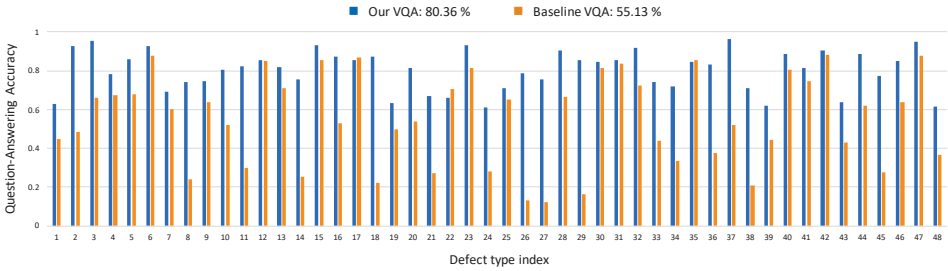

Fig. 10. The question-answering accuracy of the baseline and our layer-attentioned VQA models with feature tensor layout representation.

TABLE I
THE TESTING ACCURACIES OF VQA MODELS

	Defect type category	A	B	C	D	E	F
	# of patterns	1372	52	9159	2411	1738	664
Baseline VQA	Testing accuracy	93.88 %	94.23 %	96.99 %	97.35 %	98.39 %	98.80 %
	Miss	20	0	101	34	13	4
	False alarm	64	3	175	30	15	4
Our VQA	Testing accuracy	98.54 %	98.08 %	98.20 %	99.34 %	99.77 %	99.85 %
	Miss	1	0	64	3	0	0
	False alarm	19	1	101	13	4	1

and our VQA model in Python 2.7 with Keras and TensorFlow. Our fully connected layers are initialized with Xavier initialization [23] and separated with a batch-normalization [24] and tanh activation layer between each other. The CNN is composed of 6 two-dimensional convolutional layers, each is initialized with Xavier initialization and separated with a batch-normalization and a Leaky Rectified Linear Unit (ReLU) activation layer between each other. We set the number of hidden units in each LSTM to 128 in our VQA model. The parameters of model training are listed as follows: The initial learning rate is set to 0.01; the learning rate decay is set to 0.001; the batch size is set to 64. The whole network is optimized by Adam optimizer on softmax cross-entropy loss. All experiments are executed on a platform with NVIDIA Tesla P100.

Because prior works do not handle many-layer patterns, we compare our layer-attentioned VQA model with the baseline VQA model. First, as shown in Table I, both VQA models achieve a high average accuracy for 6 defect type categories. The testing time per pattern varies from 1 to 3 seconds. Our VQA model outperforms the baseline VQA model with higher accuracy, lower miss, and lower false alarm in every category.

Second, Fig. 10 shows the question-answering accuracy of both VQA models. Overall, the question-answering accuracy of the baseline VQA model and our VQA model is 55.13% and 80.36%, respectively. For the baseline VQA model, though it can identify whether a pattern is a hotspot, the ability of question answering to defect types is little higher than the probability of coin toss. The baseline VQA model cannot fully capture the relation between defect types and patterns to make effective predictions. Experimental results reveal our VQA model outperforms the baseline VQA model in both accuracy and question-answering accuracy.

Fig. 11 visualizes the layered relevance weights (normalizing each layer index) between defect types. Similar hotspot types have similar weight distributions. Our results reveal the layer-attentioned mechanism decides the importance of each layout layer by the defect type description, which helps the VQA model predict the critical pattern.

V. CONCLUSION

In this paper, we investigate the linkage between many-layer hotspot patterns and potentially induced defect types. We first model the many-layer critical hotspot pattern extraction task as a VQA problem: Considering a many-layer layout pattern as an image and a defect type as a question, the proposed model can answer if the pattern is critical to the queried defect type. We further devise a layer-attentioned VQA model to identify the importance and relevance of each layer. Experimental results show that the proposed model has superior question-answering ability, and the relevance of each layer varies for different defect types.

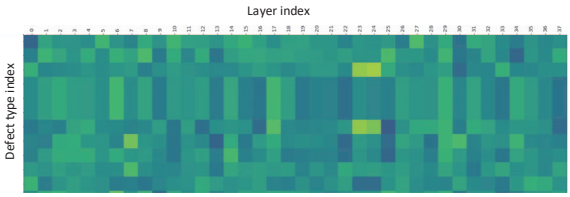

Fig. 11. The layered relevance weights of partial defect types.

REFERENCES

[1] Y. Ma et al., "Machine learning based wafer defect detection," in *SPIE Advanced Lithography, Design-Process-Technology Co-optimization for Manufacturability XIII*, vol. 10962, 2019, pp. 38 – 45.
[2] L. Francisco et al., "Multilayer CMP hotspot modeling through deep learning," in *SPIE Advanced Lithography, Design-Process-Technology Co-optimization for Manufacturability XIII*, vol. 10962, 2019, pp. 202 – 213.
[3] P.-R. Chen et al., "Systems and methods for systematic physical failure analysis (PFA) fault localization," *US Patent Application, US20200133959A1*, 2020.
[4] Y.-T. Yu et al., "Machine-learning-based hotspot detection using topological classification and critical feature extraction," *IEEE TCAD*, vol. 34, no. 3, pp. 460–470, 2015.
[5] W. C. J. Tam and R. D. S. Blanton, "LASIC: Layout analysis for systematic ic-defect identification using clustering," *IEEE TCAD*, vol. 34, no. 8, pp. 1278–1290, 2015.
[6] Y. Tomioka et al., "Lithography hotspot detection by two-stage cascade classifier using histogram of oriented light propagation," in *ASP-DAC*, 2017, pp. 81–86.
[7] R. Chen et al., "Faster region-based hotspot detection," in *DAC*, 2019, pp. 1–6.
[8] Y. Jiang et al., "Efficient layout hotspot detection via binarized residual neural network," in *DAC*, 2019, pp. 1–6.
[9] H. Yang et al., "Layout hotspot detection with feature tensor generation and deep biased learning," *IEEE TCAD*, vol. 38, no. 6, pp. 1175–1187, 2019.
[10] H. Y. et al., "Detecting multi-layer layout hotspots with adaptive squish patterns," in *ASP-DAC*, 2019, pp. 299–304.
[11] W.-C. Chang et al., "iClaire: A fast and general layout pattern classification algorithm," in *DAC*, 2017, pp. 1–6.
[12] K. Chen et al., "Minimizing cluster number with clip shifting in hotspot pattern classification," in *DAC*, 2017, pp. 1–6.
[13] M. Woo et al., "Grasp based metaheuristics for layout pattern classification," in *ICCAD*, 2017, pp. 512–518.
[14] F. Yang et al., "Improved tangent space-based distance metric for lithographic hotspot classification," *IEEE TCAD*, vol. 36, no. 9, pp. 1545–1556, 2017.
[15] S. Antol et al., "VQA: Visual question answering," in *ICCV*, 2015.
[16] T. Mikolov et al., "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013.
[17] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *ICML*, 2014.
[18] B. Zhou et al., "Simple baseline for visual question answering," *CoRR*, vol. abs/1512.02167, 2015.
[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[20] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE TSP*, vol. 45, no. 11, pp. 2673–2681, 1997.
[21] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *JMLR*, vol. 9, pp. 2579–2605, 2008.
[22] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE TKDE*, vol. 22, pp. 1345–1359, 2010.
[23] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, 2010.
[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.