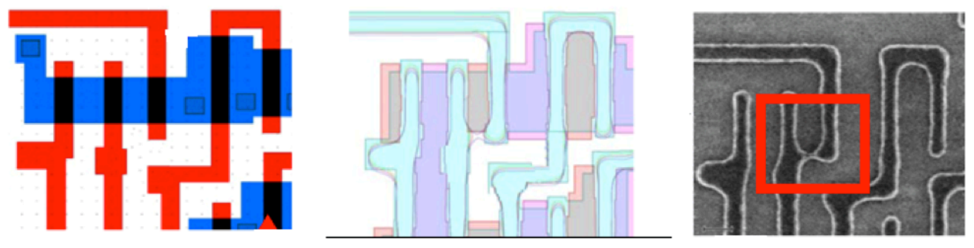
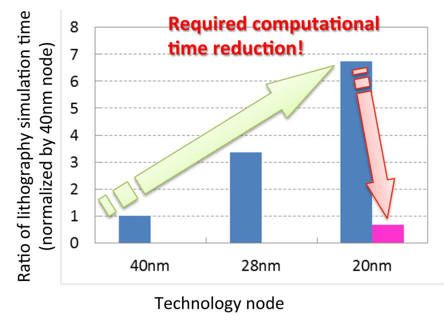
Layout Hotspot Detection with Feature Tensor Generation and Deep Biased Learning

Haoyu Yang[1], Jing Su[2], Yi Zou[2], Bei Yu[1], Evangeline F. Y. Young[1]
[1]CSE Department, Chinese University of Hong Kong, Shatin, Hong Kong
[2]ASML Brion Inc., CA 95054, USA

Introduction



- RET: OPC, SRAF, MPL
- Still hotspot: low fidelity patterns
- Simulations: extremely CPU intensive
- Hotspot Detection: Predicting patterns or regions with low printability



Preliminaries and Related Works

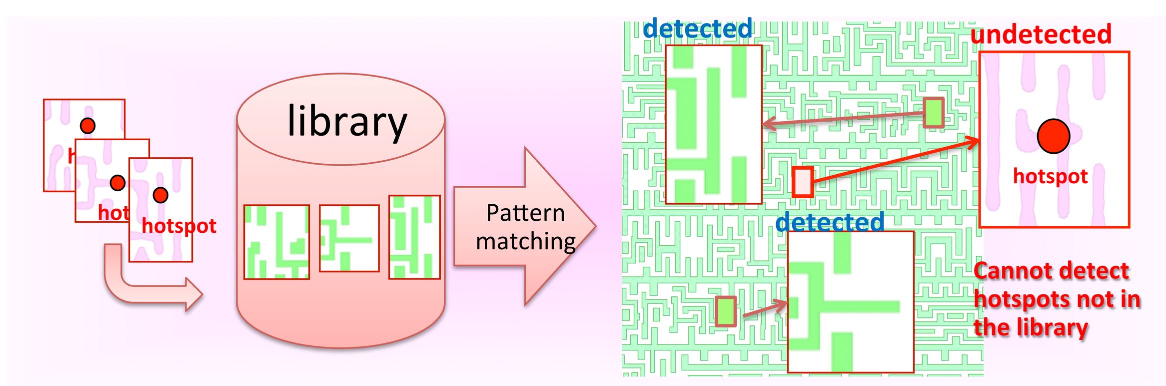
Accuracy

The ratio between the number of correctly predicted hotspot clips and the number of all real hotspot clips.

False Alarm

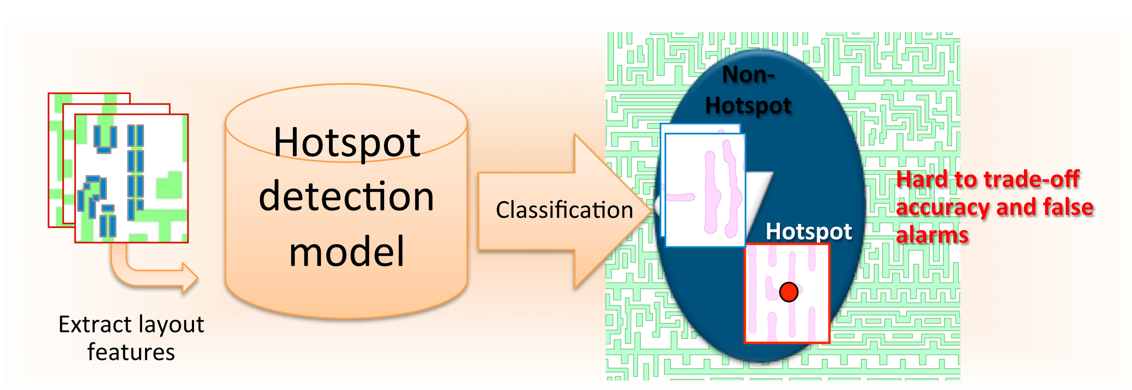
The number of non-hotspot clips that are predicted as hotspots by the classifier.

Pattern matching based hotspot detection



- Fast and accurate
- [Yu+,ICCAD'14] [Nosato+,JM3'14] [Su+,TCAD'15]
- Fuzzy pattern matching [Wen+,TCAD'14]
- Hard to detect non-seen pattern

Machine learning based hotspot detection



- Predict new patterns
- Decision-tree, ANN, SVM, Boosting, Bayesian, ...
- [Ding+,TCAD'12][Yu+,JM3'15][Matsunawa+,SPIE'15][Yu+,TCAD'15][Zhang+,ICCAD'16][Wen+,TCAD'14]
- Feature reliability and model scalability

Why Deep Learning?

1. **Feature Crafting v.s. Feature Learning**
   - Manually designed feature→Inevitable information loss
   - Learned feature→Reliable
2. **Scalability**
   - More pattern types
   - More complicated patterns
   - Hard to fit millions of data with simple ML model
3. **Mature Libraries**
   - `Caffe` [Jia+,ACMMM'14]
   - `Tensorflow` [Martin+,TR'15]

Deep Learning Issues on Hotspot Detection

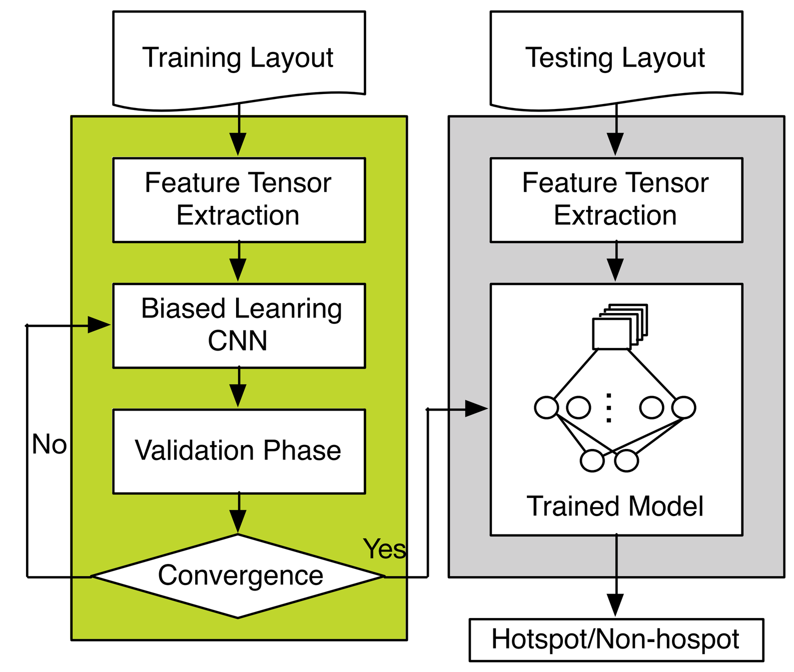
Layout image size is large ($\approx 1000 \times 1000$)

- Compared to ImageNet ($\approx 200 \times 200$)
- Associated CNN model is large
- Not storage and computational efficient

Hotspot detection accuracy is more important

- Hotspot → Circuit Failure
- False Alarm → Runtime Overhead
- Consider methods for better trade-off between accuracy and falsealarm

The Overall Detection Flow



Define the Convergence

We refer the convergence as the state when the neural network achieves satisfactory performance on the validation set.

Feature Tensor Extraction

Feature Extraction

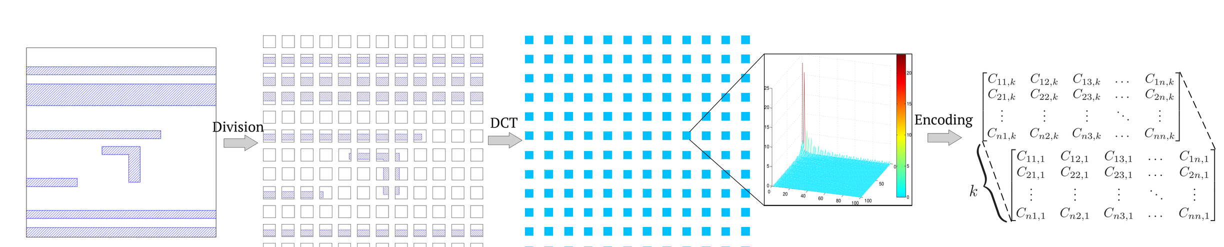
- Dimension reduction to speedup hotspot detection flow
- Density-based feature: Local pattern density affects the layout attribute
- CCS-based feature: Include prior knowledge of lithography
- Flatten into 1-D vector
- Spatial information loss

The Feature Tensor

Because the spatial relationships of mask layout patterns are important to determine the hotspot regions, it is necessary to consider layout features with higher dimension.

Extraction Procedure

1. Clip Partition
2. Discrete Cosine Transform
$$\mathbf{D}_{i,j}(m,n)=\sum_{x=0}^{B}\sum_{y=0}^{B}\mathbf{I}_{i,j}(x,y)\cos[\frac{\pi}{B}(x+\frac{1}{2})m]\cos[\frac{\pi}{B}(y+\frac{1}{2})n]$$
3. Flatten DCT coefficients
$$\mathbf{C}^*_{i,j}=[\mathbf{D}_{i,j}(0,0),\mathbf{D}_{i,j}(0,1),\mathbf{D}_{i,j}(1,0),...,\mathbf{D}_{i,j}(B,B)]^\intercal$$
4. Discarding High Frequency Components
$$\mathbf{C}_{i,j}=\mathbf{C}^*_{i,j}[:k]$$
5. Feature Tensor
$$\mathbf{F}=\begin{bmatrix}\mathbf{C}_{11} & \mathbf{C}_{12} & \mathbf{C}_{13} & \dots & \mathbf{C}_{1n}\\ \mathbf{C}_{21} & \mathbf{C}_{22} & \mathbf{C}_{23} & \dots & \mathbf{C}_{2n}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ \mathbf{C}_{n1} & \mathbf{C}_{n2} & \mathbf{C}_{n3} & \dots & \mathbf{C}_{nn}\end{bmatrix}$$



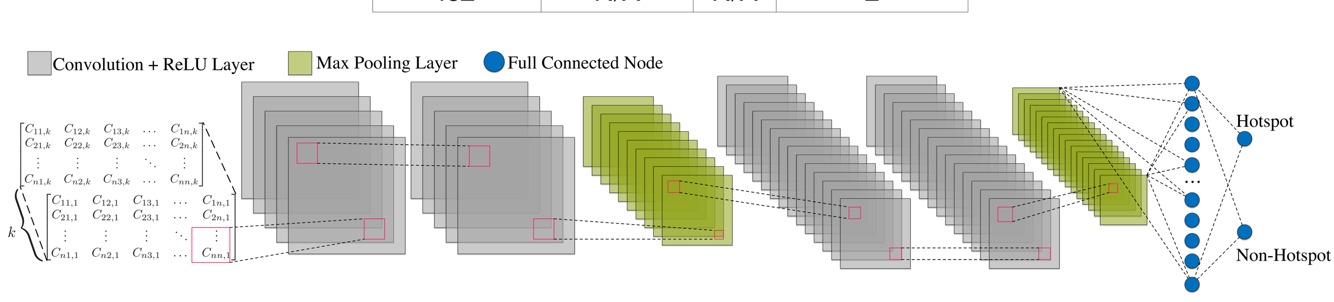
Feature Tensor Properties

- k-channel hyper-image
- Compatible with CNN
- Storage and computional efficiency

The Architecture

The CNN contains two convolution stages and two fully connected layers. Each convolution stage consists of two convolution layers and one max pooling layer. 50% dropout is applied in fc1 during training.

Layer	Kernel Size	Stride	Output Node #
conv1-1	3	1	$12 \times 12 \times 16$
conv1-2	3	1	$12 \times 12 \times 16$
maxpooling1	2	2	$6 \times 6 \times 16$
conv2-1	3	1	$6 \times 6 \times 32$
conv2-2	3	1	$6 \times 6 \times 32$
maxpooling2	2	2	$3 \times 3 \times 32$
fc1	N/A	N/A	250
fc2	N/A	N/A	2



Biased Learning Algorithm

Recall the training procedure

- Minimize difference with ground truths
$$\mathbf{y}^*_n=[1,0],\ \mathbf{y}^*_h=[0,1].$$
$$\mathbf{F}\in\begin{cases}\mathcal{N}, \text{ if } \mathbf{y}(0)>0.5\\ \mathcal{H}, \text{ if } \mathbf{y}(1)>0.5\end{cases}$$

Solutions to increase the detection accuracy

1. Shifting decision boundary
$$\mathbf{F}\in\begin{cases}\mathcal{N}, \text{ if } \mathbf{y}(0)>0.5+\lambda\\ \mathcal{H}, \text{ if } \mathbf{y}(1)>0.5-\lambda\end{cases}$$
   - Straightforward
   - At the cost of much false alarm penalties
2. Biased ground truth
$$\mathbf{y}^*_n=[1-\epsilon,\epsilon]$$
   - Sacrifice non-hotspot loss
   - Reduce the prediction score of the non-hotspot samples when they are greater than 1-ϵ
   - Do not affect the samples that are close to the decision boundary
   - Expect to have minor false alarm penalties

Assumption of the Biased Ground Truth

Given a trained convolutional neural network with ground truth $\mathbf{y}^*_n=[1,0]$ and $\mathbf{y}^*_h=[0,1]$ and hotspot detection accuracy a on a given test set. Fine tune the network with $\mathbf{y}^\epsilon_n=[1-\epsilon,\epsilon]$, $\epsilon\in[0,0.5)$, and obtain the hotspot detection accuracy a' of the new model. We have $a'\geq a$.

The training procedure

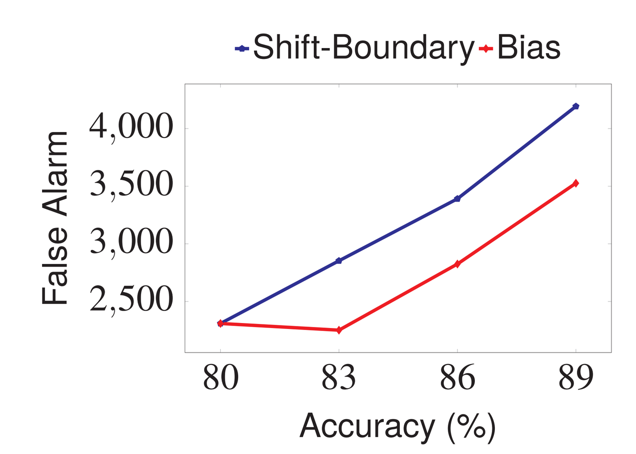
Algorithm: Biased Learning

Input: $\{\mathbf{F}_t\}, \{\mathbf{F}_v\}, \epsilon, \delta\epsilon, t, \mathbf{W}, \lambda, \alpha, k, \mathbf{y}^*_h, \mathbf{y}^*_n$;

1: $i \leftarrow 0, \epsilon \leftarrow 0, \mathbf{y}^*_h \leftarrow [0,1], \mathbf{y}^*_n \leftarrow [1-\epsilon,\epsilon]$;
2: **if** $i < t$ **then**
3: $\quad f_\epsilon \leftarrow \mathrm{MGD}(\mathbf{W}, \lambda, \alpha, k, \mathbf{y}^*_n, \mathbf{y}^*_h)$;
4: $\quad i \leftarrow i+1, \epsilon \leftarrow \epsilon+\delta\epsilon$;
5: **end if**

Effectiveness of the Biased Learning Algorithm

- The above neural network is trained with $\epsilon=0$ to obtain an initial model, and is fine-tuned with $\epsilon=0.1, 0.2, 0.3$ on `Industry3`. Then we perform boundary shifting on initial model to achieve the same test accuracy with three fine-tuned models.



Experimental Results

- Using `Python` on Intel Xeon Platform with Nvidia K620 Graphic card.
- Based on `Tensorflow` library
- Benchmarks from ICCAD Contest 2012 and Industry

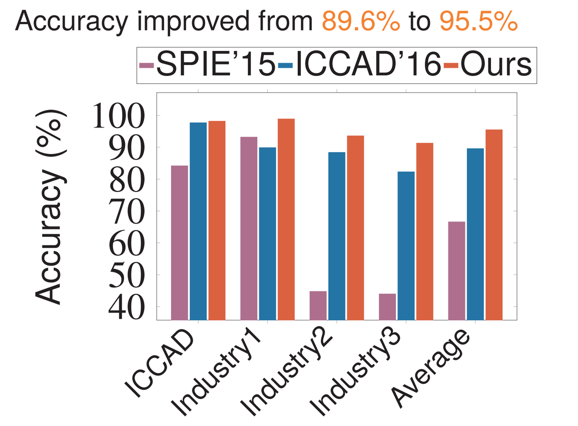
Benchmark Statistics

Benchmarks	Training Set		Testing Set	
	HS#	NHS#	HS#	NHS#
`ICCAD`	1204	17096	2524	13503
`Industry1`	34281	15635	17157	7801
`Industry2`	15197	48758	7520	24457
`Industry3`	24776	49315	12228	24817

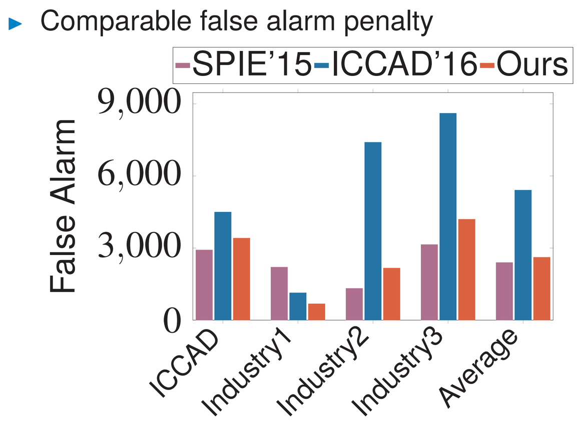
- `ICCAD` contains all the 28nm clips in the original contest benchmark
- `Industry1`–`Industry3` are from industry design and correspond to difference OPC level

Result comparison with two state-of-the-art hotspot detectors

- Accuracy improved from 89.6% to 95.5%



- Comparable false alarm penalty



Conclusions

- Propose the feature tensor representation of layout clips
- Propose the biased learning algorithm
- Demonstrate the feasibility of deep learning solutions for advanced DFM research