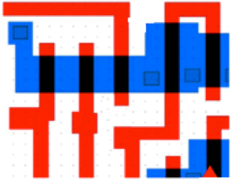
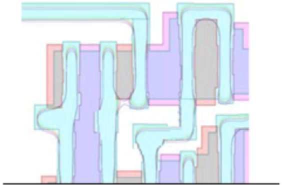
Layout Hotspot Detection with Feature Tensor Generation and Deep Biased Learning

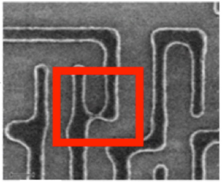
Haoyu Yang,

The Chinese University of Hong Kong

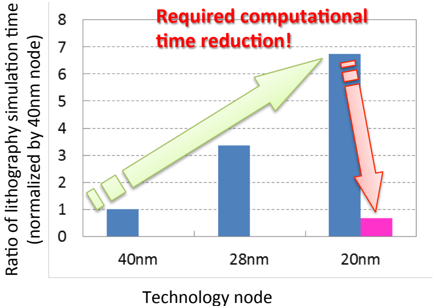
Lithography Hotspot Detection







- RET: OPC, SRAF, MPL
- Still hotspot: low fidelity patterns
- Simulations: extremely CPU intensive



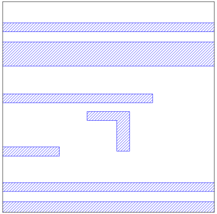
Special Issues for Layout Hotspot Detection

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

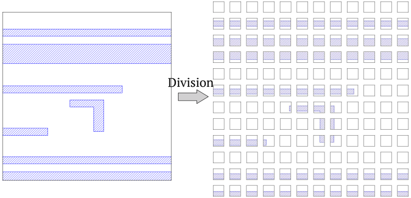
- Hotspot –> Circuit Failure
- False Alarm –> Runtime Overhead
- Consider methods for better trade-off between accuracy and falsealarm



Layout clip with $1nm$ precision has resolution 1200×1200

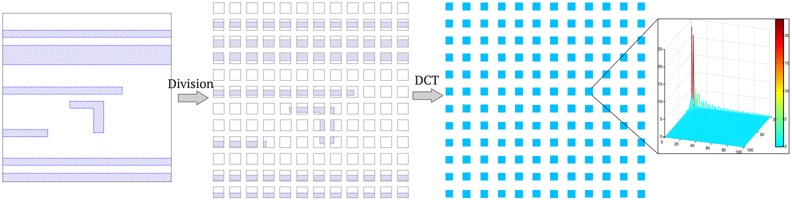
Feature Tensor Generation

- Clip Partition
- Discrete Cosine Transform
- Discarding High Frequency Components
- Feature Tensor



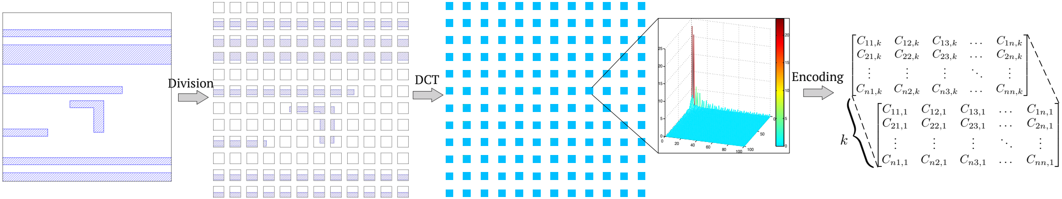
Feature Tensor Generation

- Clip Partition
- Discrete Cosine Transform
- Discarding High Frequency Components
- Feature Tensor



Feature Tensor Generation

- Clip Partition
- Discrete Cosine Transform
- Discarding High Frequency Components
- Feature Tensor

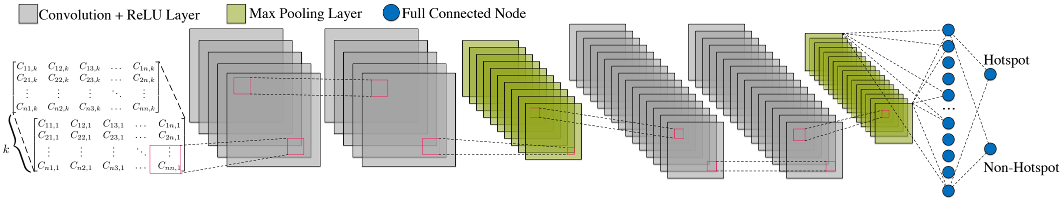


CNN Architecture

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

Feature Tensor

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



Recall The Training Procedure

- Minimize difference with ground truths

$$\mathbf{y}_n^* = [1, 0], \ \mathbf{y}_h^* = [0, 1]. \tag{1}$$

$$\mathbf{F} \in \begin{cases} \mathcal{N}, & \text{if } \mathbf{y}(0) > 0.5 \\ \mathcal{H}, & \text{if } \mathbf{y}(1) > 0.5 \end{cases} \tag{2}$$

Recall The Training Procedure

- Minimize difference with ground truths

$$\mathbf{y}_n^* = [1,0],\ \mathbf{y}_h^* = [0,1]. \tag{1}$$

$$\mathbf{F} \in \begin{cases} \mathcal{N}, & \text{if } \mathbf{y}(0) > 0.5 \\ \mathcal{H}, & \text{if } \mathbf{y}(1) > 0.5 \end{cases} \tag{2}$$

- Shifting decision boundary

$$\mathbf{F} \in \begin{cases} \mathcal{N}, & \text{if } \mathbf{y}(0) > 0.5 + \lambda \\ \mathcal{H}, & \text{if } \mathbf{y}(1) > 0.5 - \lambda \end{cases} \tag{3}$$

Recall The Training Procedure

- Minimize difference with ground truths

$$\mathbf{y}_n^* = [1, 0], \ \mathbf{y}_h^* = [0, 1]. \tag{1}$$

$$\mathbf{F} \in \begin{cases} \mathcal{N}, & \text{if } \mathbf{y}(0) > 0.5 \\ \mathcal{H}, & \text{if } \mathbf{y}(1) > 0.5 \end{cases} \tag{2}$$

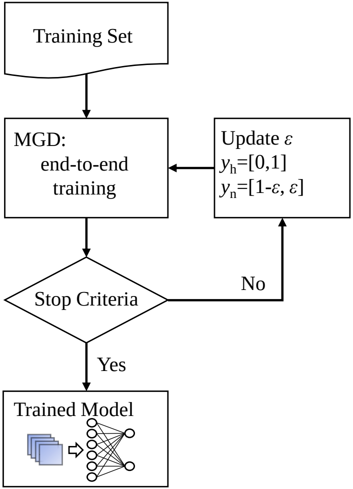
- Shifting decision boundary (✗)

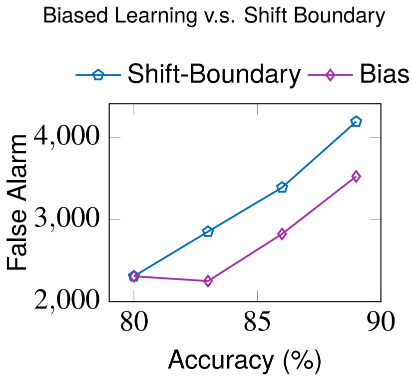
$$\mathbf{F} \in \begin{cases} \mathcal{N}, & \text{if } \mathbf{y}(0) > 0.5 + \lambda \\ \mathcal{H}, & \text{if } \mathbf{y}(1) > 0.5 - \lambda \end{cases} \tag{3}$$

- Biased ground truth

$$\mathbf{y}_n^* = [1 - \epsilon, \epsilon] \tag{4}$$

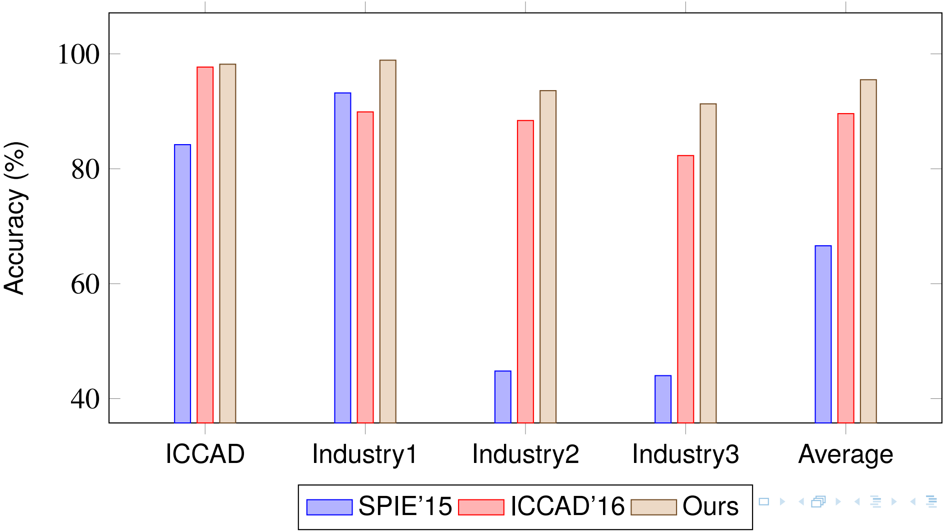
The Biased Learning Algorithm





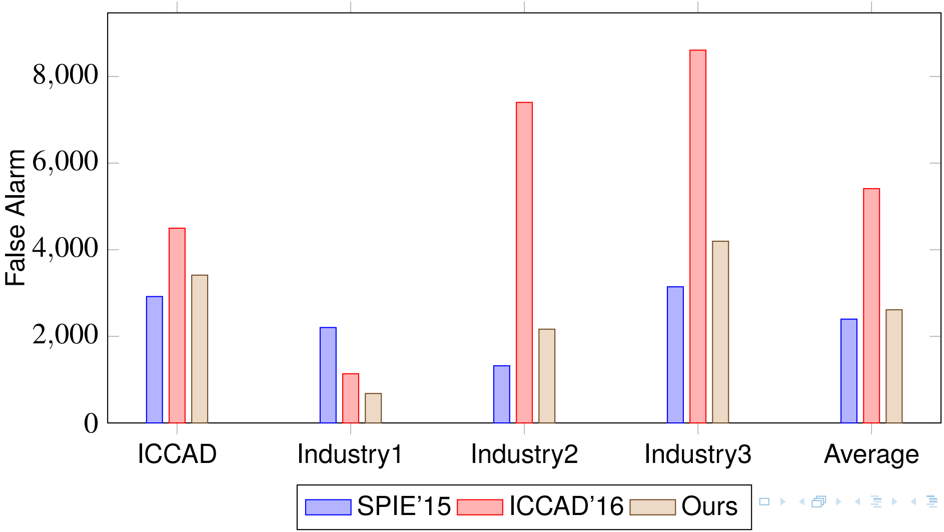
Comparison with Two Hotspot Detectors

- Detection accuracy improved from 89.6% to 95.5%



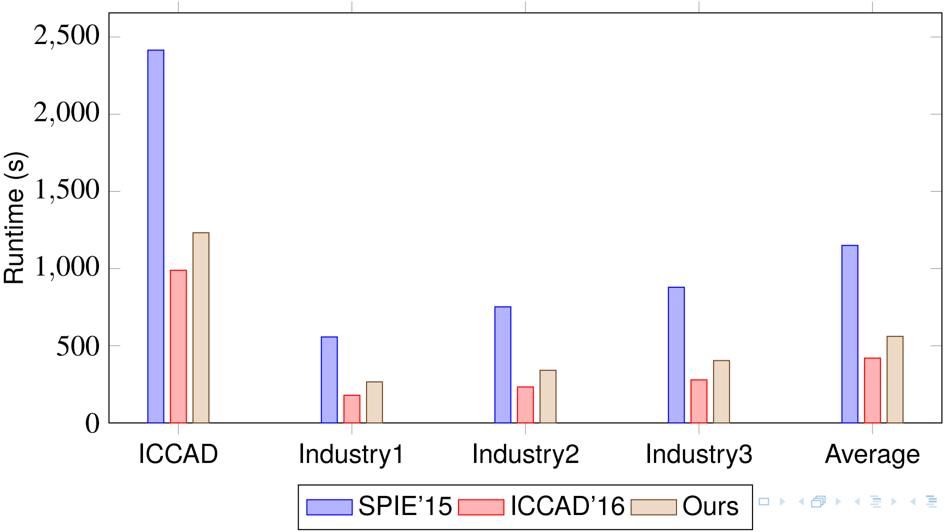
Comparison with Two Hotspot Detectors

- Comparable false alarm penalty



Comparison with Two Hotspot Detectors

- Comparable testing runtime



Thank You