# Hotspot Detection via Multi-task Learning and Transformer Encoder



**Binwu Zhu**[1], Ran Chen[1], Xinyun Zhang[1], Fan Yang[2], Xuan Zeng[2], Bei Yu[1], Martin D.F. Wong[1]

[1]The Chinese University of Hong Kong
[2]Fudan University
{bwzhu,byu}@cse.cuhk.edu.hk

Nov. 1, 2021

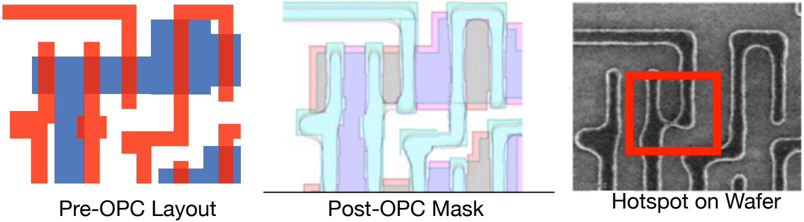
# Outline

1. Introduction
2. Method
3. Results

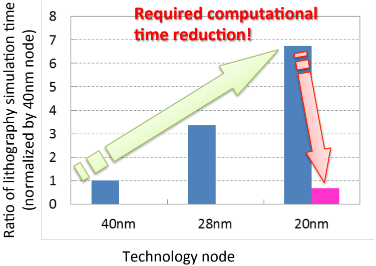
# Outline

1. Introduction
2. Method
3. Results

# Lithography Hotspot Detection




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

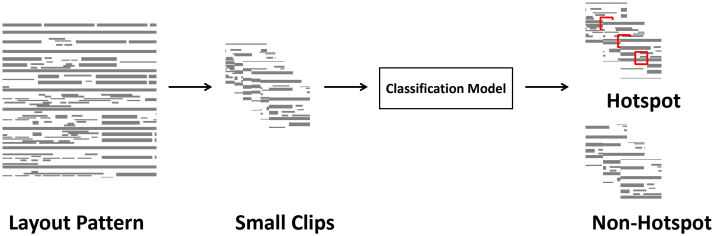


# Hotspot Detection Methods

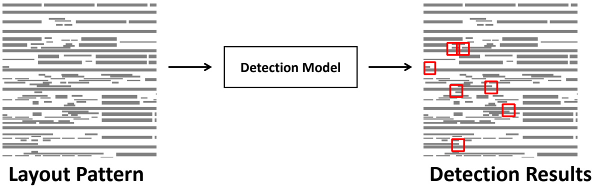
- **Pattern Matching**
    - Characterize the hotspots as explicit patterns and identify the hotspots by matching these patterns.
    - **Drawback**: Fast but hard to detect unseen patterns.
- **Machine Learning**
    - Build implicit models by learning from existing data, which can detect unseen hotspots.
    - A trade-off between accuracy and efficiency.

# Two Perspectives of Machine Learning Based Solutions

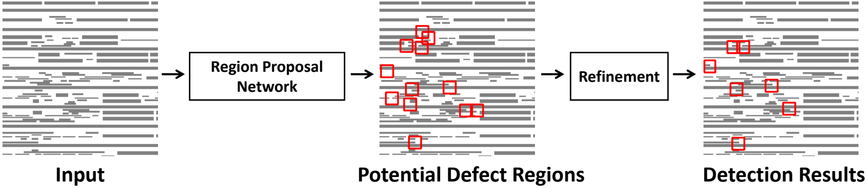
- Classification Perspective




- Detection Perspective



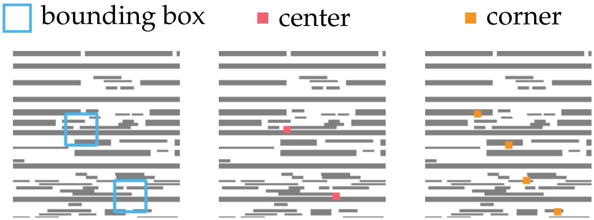
- Methods from classification perspective is still time-consuming.
- R-HSD[1] is from detection perspective but the whole flow requires two stages.




[1] Ran Chen et al. (2019). "Faster Region-based Hotspot Detection". In: *DAC*, pp. 1–6.

# Our Motivation - II

- Underlying information, e.g. corner information, has not been utilized in previous work, which is also helpful for detection tasks in some aspects.
- Corner information[2] contributes to improve the localization accuracy, while center information[3] performs better on detecting small targets.

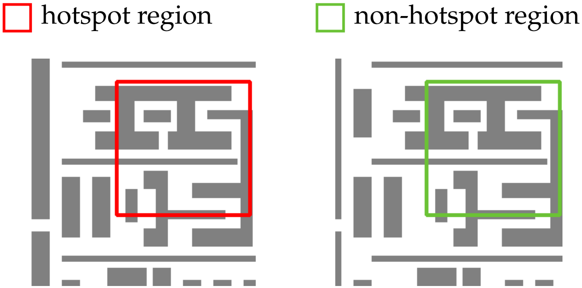



[2]Hei Law and Jia Deng (2018). "Cornernet: Detecting objects as paired keypoints". In: *ECCV*, pp. 734–750.

[3]Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl (2019). "Objects as points". In: *arXiv preprint arXiv:1904.07850*.

# Our Motivation - III

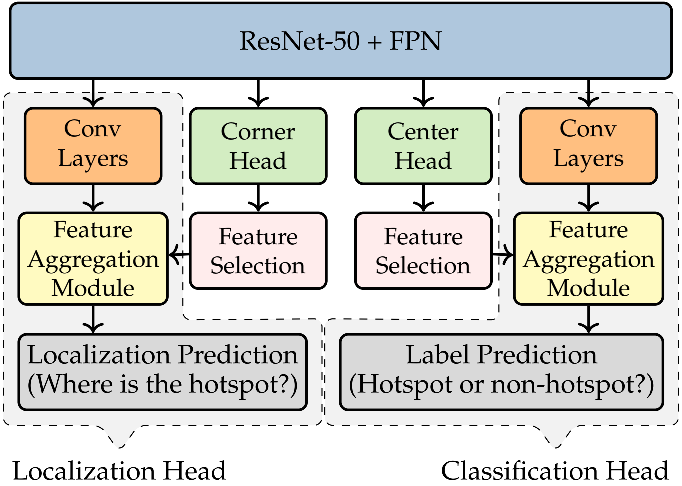
- We observe a strange situation where two regions sharing the same layout patterns may have different simulation results.




- CNNs, commonly adopted by previous methods, are infeasible to capture the long-range dependencies due to the locality inductive bias.

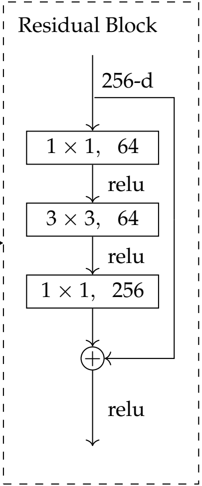
# Outline

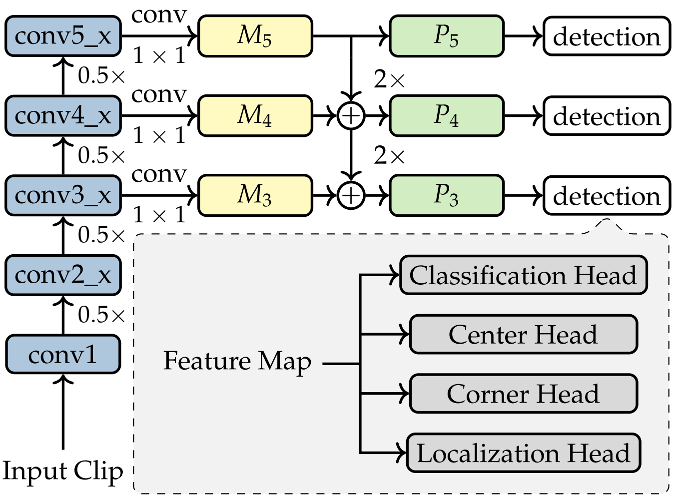
# Overall Framework



# Backbone – ResNet-50

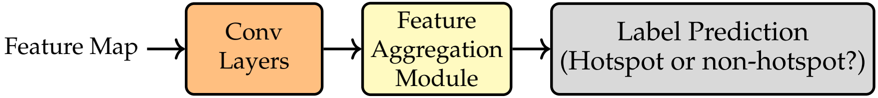
| layer name | ResNet-50 |
|---|---|
| conv1 | $7 \times 7, 64$, stride 2 |
| conv2_x | $3 \times 3$ max pool, stride 2<br>$\left[\begin{array}{c} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{array}\right] \times 3$ |
| conv3_x | $\left[\begin{array}{c} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{array}\right] \times 4$ |
| conv4_x | $\left[\begin{array}{c} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{array}\right] \times 6$ |
| conv5_x | $\left[\begin{array}{c} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{array}\right] \times 3$ |






- The width $w_k$ of the feature map $P_k(M_k)$ is equal to $w_i/2^k$, where $w_i$ is the width of the input.

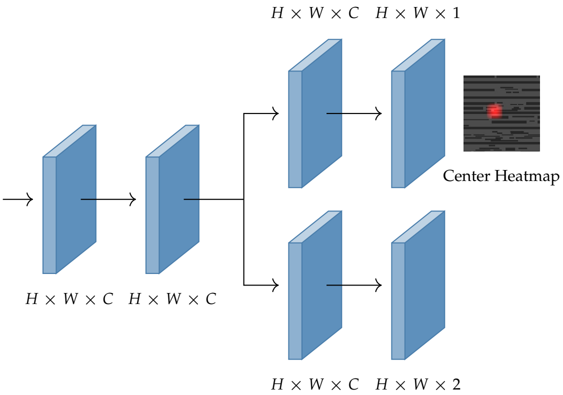
# Classification and Localization Head




- Define $M$ anchors with different sizes and ratios for each pixel of the feature map $F \in \mathbb{R}^{H \times W \times C}$.
- Feature aggregation module is adopted to enhance the feature map.
- The last layer predicts the label of each anchor and output $F_o \in \mathbb{R}^{H \times W \times M}$.

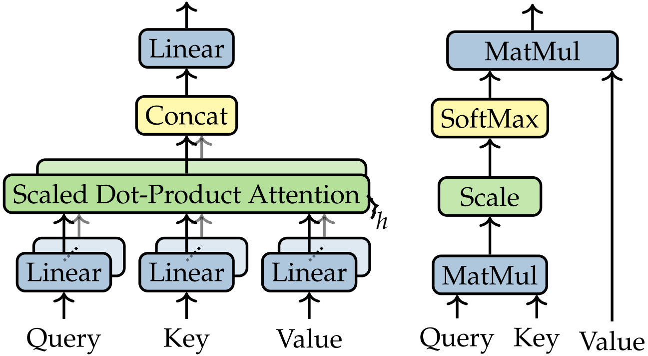
# Center and Corner Head

- **Inspiration**: By jointly training the hotspot detector to learn different but related tasks, the knowledge learned from one task can be leveraged by others.
- Center Head Structure



# Feature Aggregation Module

- **Preliminary**: Transformer and Multi-Head Attention




- $\mathrm{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{Concat}\left(\mathbf{H}_1, \ldots, \mathbf{H}_h\right)\mathbf{W}^O$, where $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{n \times d_m}$
- $\mathbf{H}_i = \mathrm{Attention}\left(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V\right) = \mathrm{softmax}\left[\frac{\mathbf{Q}\mathbf{W}_i^Q\left(\mathbf{K}\mathbf{W}_i^K\right)^\top}{\sqrt{d_k}}\right]\mathbf{V}\mathbf{W}_i^V.$

# Feature Aggregation Module

- **Objective**: Globally capture the dependencies between different features in the feature map output by the Conv Layers module in Localization and Classification Head (Motivation III).
- **Inspiration**: The Transformer provides a powerful way in modeling all pairwise interactions between any two features.
- But what will happen if we simply adopt Multi-Head Attention by augmenting $f_m \in \mathbb{R}^C$ with other features? The output can be formulated as follows:

$$\boldsymbol{h}_i = \sum_{n=1}^{HW} \frac{\exp(\boldsymbol{f}_m \boldsymbol{W}_i^Q (\boldsymbol{f}_n \boldsymbol{W}_i^K)^\top / \sqrt{C})}{\sum_{n=1}^{HW} \exp(\mathbf{f}_m \boldsymbol{W}_i^Q (\boldsymbol{f}_n \boldsymbol{W}_i^K)^\top / \sqrt{C})} \boldsymbol{f}_n \boldsymbol{W}_i^V$$

- **Drawback**: Expensive computation cost with complexity $\mathcal{O}(HWC^2)$.

# Feature Aggregation Module

- Based on Transformer, we need some methods to **save the computation cost**.
- **Overview**: Leverage the knowledge from the Center Head and Corner Head to **guide the key selection** for the feature map in the Classification Head and Localization Head separately.

**Algorithm 1** Key selection algorithm

**Input:** Center Probability Map $\boldsymbol{C} \in \mathbb{R}^{H \times W}$, Feature Map $\boldsymbol{F}_c \in \mathbb{R}^{H \times W \times C}$, Selection Number $k$;
**Output:** Key set $\boldsymbol{F}_k$;

```
1: F_k ← Initialized to empty set;
2: C′ ← AvgPool(C);
3: topk_idx ← the index of the maximum k values in C′;
4: for i ← 1, 2, ... k do
5:     idx ← topk_idx[i];
6:     f_i ← the feature in the position idx of F_c;
7:     append feature f_i to key set F_k;
8: end for
9: return Key set F_k with k features.
```

- $h_i = f_m + \lambda \sum_{n=1}^{k} \frac{\exp(f_m W_i^Q (f_n W_i^K)^\top / \sqrt{C})}{\sum_{n=1}^{k} \exp(f_m W_i^Q (f_n W_i^K)^\top / \sqrt{C})} f_n W_i^V$. The complexity of the optimized algorithm decreases to $\mathcal{O}(kC^2)$.

# Outline

- Benchmarks from ICCAD Contest 2016.
- "Accu(%)": Accuracy, the ratio between the number of correctly categorized hotspot clips and the number of real hotspot clips.
- "FA": False Alarm, non-hotspot clips that are classified as hotspots

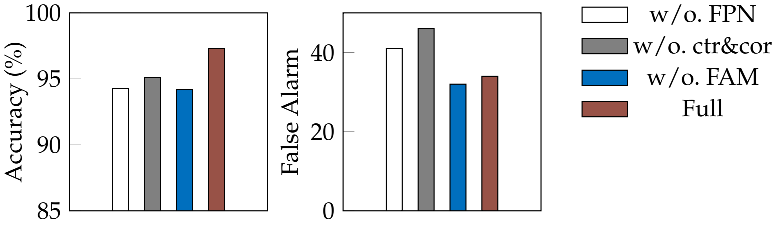
| Bench | TCAD'19[4] | | | R-HSD[5] | | | Ours | | |
|---|---|---|---|---|---|---|---|---|---|
| | Accu(%) | FA | Time(s) | Accu(%) | FA | Time(s) | Accu(%) | FA | Time(s) |
| `case2` | 77.78 | 48 | 60.0 | 93.02 | 17 | 2.0 | 94.87 | 6 | 1.0 |
| `case3` | 91.20 | 263 | 265.0 | 94.5 | 34 | 10.0 | 97.2 | 26 | 4.0 |
| `case4` | 100.00 | 511 | 428.0 | 100.0 | 201 | 6.0 | 100 | 70 | 6.0 |
| Average | 89.66 | 274.00 | 251.00 | 95.84 | 84.00 | 6.00 | **97.31** | **34.00** | **3.67** |
| Ratio | 0.92 | 8.06 | 67.84 | 0.98 | 2.47 | 1.62 | 1.00 | 1.00 | 1.00 |

[4] Haoyu Yang et al. (2019). "Layout hotspot detection with feature tensor generation and deep biased learning". In: *TCAD*. vol. 38. 6. IEEE, pp. 1175–1187.

[5] Ran Chen et al. (2019). "Faster Region-based Hotspot Detection". In: *DAC*, pp. 1–6.

# Ablation Study

- "w/o. FPN": the detector without Feature Pyramid Network
- "w/o. ctr&cor": the detector without Center Head and Corner Head
- "w/o. FAM": the detector trained without Feature Aggregation Module
- "Ful": the proposed detector




**(a)** **(b)**

# THANK YOU!