RouteNet: Routability Prediction for Mixed-Size Designs Using Convolutional Neural Network

Zhiyao Xie¹, Yu-Hung Huang², Guan-Qi Fang², Haoxing Ren³, Shao-Yun Fang², Yiran Chen¹, Jiang Hu⁴

¹Duke University, Durham, NC, USA
²National Taiwan University of Science and Technology, Taipei, TW
³Nvidia Corporation, Austin, TX, USA
⁴Texas A&M University, College Station, TX, USA

ICCAD 2018
Background

• Challenges in Routability Prediction:
  • Predict routability at placement stage
    • Predict by ‘fast trial global routing’
    • Not fast enough
  • Predict locations of Design Rule Checking (DRC) hotspots
    • Predict by global routing
    • Not accurate enough
Background

• Our Attempt on Such Challenges:
  • Fast routability forecast for placement
    • In terms of number of Design Rule Violations (#DRV)
    • To identify more routable placements among many candidates
  • Prediction of DRC hotspot locations
    • To proactively modify solutions to prevent design rule violations
Background

• Previous solutions:
  • Many fail to consider macros
  • Some require Global Routing information for #DRV prediction

<table>
<thead>
<tr>
<th>Methods</th>
<th>Use GR?</th>
<th>Predict #DRV?</th>
<th>Predict hotspot?</th>
<th>Handle macros?</th>
</tr>
</thead>
<tbody>
<tr>
<td>[18] (Qi, et al., ICCD14)</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>[26] (Zhou, et al., ASQED15)</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>[3] (Chan, et al., ICCD16)</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>[4] (Chan, et al., ISPD17)</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>RouteNet #DRV prediction</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>RouteNet hotspot prediction</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

GR means global routing
#DRV means number of DRC violations
Background

• Previous solutions:
  • Many apply machine learning on every *small cropped region*
Background

• Challenges of Macros on #DRV prediction:
  • Correlation between pin density and #DRV largely disappears with macro

Correlation between #DRV and coefficient of variation ($\frac{\sigma}{\mu}$) of pin density. Each point corresponds to one placement.
Background

• Challenges of Macros on hotspot detection:
  • Hotspots tend to aggregate at small gaps between neighboring macros
Background

• Challenges of Macros:
  • A layout with macros is much less homogeneous
  • **Homogeneity** implies resemblance among different regions of layout
  • Need a larger region to capture global view
  • **Use deep neural network!**
Background

- Convolutional Neural Network (CNN)
  - Convolutional (CONV), Pooling (POOL) and Fully Connected (FC) layers
  - Widely used in image classification

- Fully Convolutional Network (FCN)
  - Eliminate FC layers
  - May use transposed-convolutional to up-sample
  - Used in image segmentation, object detection

Features Extraction

- **RUDY** (Rectangular Uniform wire DensitY) (P. Spinder et al. 2007)
  - RUDY is a pre-routing congestion estimator
  - At \((x, y)\), for \(k\) th net with bounding box \(\{x_{min}^k, x_{max}^k, y_{min}^k, y_{max}^k\}\):
    \[
    w^k = x_{max}^k - x_{min}^k, \quad h^k = y_{max}^k - y_{min}^k
    \]
    \[
    c^k = \begin{cases} 
    1 & x \in [x_{min}^k, x_{max}^k], \quad y \in [y_{min}^k, y_{max}^k] \\
    0 & \text{otherwise} 
    \end{cases}
    \]
    \[
    RUDY^k(x, y) \propto c^k \frac{w^k + h^k}{w^k \times h^k}
    \]
    \[
    RUDY(x, y) = \sum_{k=1}^{K} RUDY^k(x, y)
    \]
Features Extraction

• $X_{ij} = j^{th}$ feature in $i^{th}$ placement

Grid size: $l \, \mu m$

- $l = \frac{H}{I} \Rightarrow h = \frac{H}{l}$
- $w = \frac{W}{l}$

$X_{ij} \in \mathbb{R}^{w \times h}$
Features Extraction

- **Macro:**
  - region occupied by macros
  - density of macro pins in each layer

- **Cell:**
  - density of cells
  - density of cell pins

- **Global cell:**
  - cell features at global placement

- **Global RUDY:**
  - RUDY features calculated by global placement results
Features Extraction

- **RUDY**
  - long-range RUDY
    - RUDY from long-range nets
  - short-range RUDY
    - DURY from short-range nets
  - RUDY pins
    - pins with density value equal to the RUDY value of its net

- **Congestion**
  - trial global routing congestion
  - global routing congestion

- **DRC violation**
  - prediction target / label

For $i^{th}$ placement with size $w \times h$ and $F$ features:

$$X_i \in \mathbb{R}^{w \times h \times F}$$
Input tensor constructed by stacking 2D features:
(1) Pin density, (2) macro (3) long-range RUDY, (4) RUDY pins

Input features for #DRV prediction.
Red: macro region
Green: global long-range RUDY
Blue: global RUDY pins
Proposed Model

Problem 1 (#DRV prediction). Find an estimator $f_{\#DRV}^*$ of DRV count in a placement:

$$f_{\#DRV} : X_i^{(#DRV)} \in \mathbb{R}^{w \times h \times F_1} \rightarrow y_i \in \mathbb{N}$$

$$f_{\#DRV}^* = \arg \min_f \text{Loss}(f(X_i^{(#DRV)}), y_i)$$

Problem 2 (Hotspot prediction). Find a detector $f_{\text{hotspot}}^*$ of hotspots. It reports locations of all DRC hotspots in a placement.

$$f_{\text{hotspot}} : X_i^{(\text{hotspot})} \in \mathbb{R}^{w \times h \times F_2} \rightarrow V_i \in \{0, 1\}^{w \times h}$$

$$f_{\text{hotspot}}^* = \arg \min_f \text{Loss}(f(X_i^{(\text{hotspot})}), V_i)$$

$$Y_i \in \mathbb{R}^{w \times h} \quad V_{imn} = 1(Y_{imn} > \epsilon)$$
Proposed Model- #DRV Prediction

Algorithm 1 Algorithm of RouteNet for #DRV Prediction

Input: Number of training placements: $N$, Features: $\{X_i \in \mathbb{R}^{w \times h \times 3} \mid i \in [1, N]\}$, Targets: $\{y_i \in \mathbb{R} \mid i \in [1, N]\}$

Preprocess:

1. for each int $i \in [1, N]$ do
2.   Resize $X_i \in \mathbb{R}^{w \times h \times 3}$ into $X_i^{#DRV} \in \mathbb{R}^{224 \times 224 \times 3}$
3. Find 25%, 50%, 75% quantizes of $y_i$: $q_1$, $q_2$, $q_3$
4. for each int $i \in [1, N]$ do
5.   $C_i \leftarrow 0$
6.   for each int $t \in [1, 3]$ do
7.     if $y_i > q_t$ then
8.       $C_i \leftarrow t$, break
9. Form dataset $\{(X_i^{#DRV}, C_i) \mid i \in [1, N]\}$
10. Training set $\{(X_i^{#DRV}, C_i) \mid C_i = 0 \text{ or } C_i = 3\}$

Training:

1. Get pretrained ResNet18 $f_{Res} : \mathbb{R}^{224 \times 224 \times 3} \rightarrow \mathbb{R}^{1000}$
2. Replace output layer, s.t. $f_{#DRV} : \mathbb{R}^{224 \times 224 \times 3} \rightarrow \mathbb{R}$
3. Choose MSE as loss function, SGD for optimization
4. Train $f_{#DRV}$ with preprocessed dataset for $\sim 30$ epochs

Output: $f_{#DRV}$ estimating #DRV level

Resize input to 224*224, to utilize models pre-trained on images with size 224*224
Proposed Model - #DRV Prediction

**Algorithm 1** Algorithm of RouteNet for #DRV Prediction

**Input:** Number of training placements: $N$, Features: $\{X_i \in \mathbb{R}^{w \times h \times 3} \mid i \in [1, N]\}$, Targets: $\{y_i \in \mathbb{R} \mid i \in [1, N]\}$

**Preprocess:**
1. for each int $i \in [1, N]$ do
2.   Resize $X_i \in \mathbb{R}^{w \times h \times 3}$ into $X_i^{\#DRV} \in \mathbb{R}^{224 \times 224 \times 3}$
3. Find 25%, 50%, 75% quantizes of $y_i$: $q_1, q_2, q_3$
4. for each int $i \in [1, N]$ do
5.   $C_i \leftarrow 0$
6.   for each int $t \in [1, 3]$ do
7.     if $y_i > q_t$ then
8.       $C_i \leftarrow t$, break
9. Form dataset $\{(X_i^{\#DRV}, C_i) \mid i \in [1, N]\}$
10. Training set $\{(X_i^{\#DRV}, C_i) \mid C_i = 0 \text{ or } C_i = 3\}$

**Training:**
1. Get pretrained ResNet18 $f_{Res} : \mathbb{R}^{224 \times 224 \times 3} \rightarrow \mathbb{R}^{1000}$
2. Replace output layer, s.t. $f_{\#DRV} : \mathbb{R}^{224 \times 224 \times 3} \rightarrow \mathbb{R}$
3. Choose MSE as loss function, SGD for optimization
4. Train $f_{\#DRV}$ with preprocessed dataset for $\sim 30$ epoches

**Output:** $f_{\#DRV}$ estimating #DRV level

Assign placements to 4 different classes ($c_0, c_1, c_2, c_3$) based on their level of violations (#DRV)

$c_0$ represents least #DRV, while $c_3$ represents most
Proposed Model - \#DRV Prediction

Algorithm 1 Algorithm of RouteNet for \#DRV Prediction

**Input:** Number of training placements: \( N \), Features: 
\( \{X_i \in \mathbb{R}^{w \times h \times 3} \mid i \in [1, N]\} \), Targets: \( \{y_i \in \mathbb{R} \mid i \in [1, N]\} \)

**Preprocess:**
1. for each int \( i \in [1, N] \) do
2.   Resize \( X_i \in \mathbb{R}^{w \times h \times 3} \) into \( X_i^{\#DRV} \in \mathbb{R}^{224 \times 224 \times 3} \)
3. Find 25\%, 50\%, 75\% quantizes of \( y_i \): \( q_1, q_2, q_3 \)
4. for each int \( i \in [1, N] \) do
5.   \( C_i \leftarrow 0 \)
6.   for each int \( t \in [1, 3] \) do
7.     if \( y_i > q_t \) then
8.       \( C_i \leftarrow t, \text{ break} \)
9. Form dataset \( \{(X_i^{\#DRV}, C_i) \mid i \in [1, N]\} \)
10. Training set \( \{(X_i^{\#DRV}, C_i) \mid C_i = 0 \text{ or } C_i = 3\} \)

**Training:**
1. Get pretrained ResNet18 \( f_{\text{Res}} : \mathbb{R}^{224 \times 224 \times 3} \rightarrow \mathbb{R}^{1000} \)
2. Replace output layer, s.t. \( f_{\#DRV} : \mathbb{R}^{224 \times 224 \times 3} \rightarrow \mathbb{R} \)
3. Choose MSE as loss function, SGD for optimization
4. Train \( f_{\#DRV} \) with preprocessed dataset for \( \sim 30 \) epochs

**Output:** \( f_{\#DRV} \) estimating \#DRV level

Download a pre-trained CNN model named ResNet18
Proposed Model - #DRV Prediction

Algorithm 1 Algorithm of RouteNet for #DRV Prediction

**Input:** Number of training placements: $N$, Features: \( X_i \in \mathbb{R}^{w \times h \times 3} \) | $i \in [1, N]$}, Targets: \( y_i \in \mathbb{R} \) | $i \in [1, N]$}

**Preprocess:**
1. **for** each int $i \in [1, N]$ **do**
2. \( \text{Resize } X_i \in \mathbb{R}^{w \times h \times 3} \text{ into } X_i^{#DRV} \in \mathbb{R}^{224 \times 224 \times 3} \)
3. Find 25%, 50%, 75% quantizes of $y_i$: $q_1$, $q_2$, $q_3$
4. **for** each int $i \in [1, N]$ **do**
5. \( C_i \leftarrow 0 \)
6. **for** each int $t \in [1, 3]$ **do**
7. \( \text{if } y_i > q_t \text{ then} \)
8. \( C_i \leftarrow t, \text{break} \)
9. Form dataset \( \{(X_i^{#DRV}, C_i) | i \in [1, N]\} \)
10. Training set \( \{(X_i^{#DRV}, C_i) | C_i = 0 \text{ or } C_i = 3\} \)

**Training:**
1. Get pretrained ResNet18 $f_{Res} : \mathbb{R}^{224 \times 224 \times 3} \rightarrow \mathbb{R}^{1000}$
2. Replace output layer, s.t. $f_{#DRV} : \mathbb{R}^{224 \times 224 \times 3} \rightarrow \mathbb{R}$
3. Choose MSE as loss function, SGD for optimization
4. Train $f_{#DRV}$ with preprocessed dataset for $\sim 30$ epochs

**Output:** $f_{#DRV}$ estimating #DRV level

Fine-tune CNN with preprocessed data
Proposed Model - Hotspot Detection

Pixel-wise loss function

\[ Y_{i,m,n}^{\text{clip}} = \min(Y_{i,m,n}, c) \]

\[ \text{Loss} = \sum_{i=1}^{N} \sum_{m=1}^{w} \sum_{n=1}^{h} \| f_{\text{hotspot}}(X_{i,m,n}) - Y_{i,m,n}^{\text{clip}} \|_2 + \lambda \| W \|_2 \]
Data

• Five designs from ISPD 2015
• ~300 different placements by placing macros in different way
• When each design tested, model trained only on four other designs
• SVM and Logistic Regression (LR) methods for comparison

<table>
<thead>
<tr>
<th>Circuit Name</th>
<th>#Macros</th>
<th>#Cells</th>
<th>#Nets</th>
<th>Width (µm)</th>
<th>#Placements</th>
</tr>
</thead>
<tbody>
<tr>
<td>des_perf</td>
<td>4</td>
<td>108288</td>
<td>110283</td>
<td>900</td>
<td>600</td>
</tr>
<tr>
<td>edit_dist</td>
<td>6</td>
<td>127413</td>
<td>131134</td>
<td>800</td>
<td>300</td>
</tr>
<tr>
<td>fft</td>
<td>6</td>
<td>30625</td>
<td>32088</td>
<td>800</td>
<td>300</td>
</tr>
<tr>
<td>matrix_mult_a</td>
<td>5</td>
<td>149650</td>
<td>154284</td>
<td>1500</td>
<td>300</td>
</tr>
<tr>
<td>matrix_mult_b</td>
<td>7</td>
<td>146435</td>
<td>151614</td>
<td>1500</td>
<td>300</td>
</tr>
</tbody>
</table>
#DRV Prediction Evaluation

- How methods recognize placements with the lowest #DRV level ($c_0$)
- The quality of placements selected by each method
  - The best rank of top ten placements predicted to have least #DRV

<table>
<thead>
<tr>
<th>Circuit Name</th>
<th>$c_0/c_1+c_2+c_3$ accuracy (%)</th>
<th>Best rank in top 10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SVM</td>
<td>LR</td>
</tr>
<tr>
<td>des_perf</td>
<td>63</td>
<td>74</td>
</tr>
<tr>
<td>edit_dist</td>
<td>69</td>
<td>68</td>
</tr>
<tr>
<td>fft</td>
<td>66</td>
<td>62</td>
</tr>
<tr>
<td>matrix_mult_a</td>
<td>66</td>
<td>65</td>
</tr>
<tr>
<td>matrix_mult_b</td>
<td>63</td>
<td>62</td>
</tr>
<tr>
<td>Average</td>
<td>65</td>
<td>66</td>
</tr>
</tbody>
</table>
#DRV Prediction Evaluation

• **Y**: gap between the ‘best in 10’ and the actually 1st-ranked placement with least #DRV

• **X**: inference time taken for each method

• RouteNet achieves low inference time and high accuracy at the same time
DRC Hotspot Detection Evaluation

- Same decision threshold is used for all designs
- Slight different FPR, but all under 1%
- RouteNet is superior to all methods and improves global routing accuracy by 50%

<table>
<thead>
<tr>
<th>Circuit Name</th>
<th>FPR (%)</th>
<th>TR</th>
<th>GR</th>
<th>LR</th>
<th>SVM</th>
<th>RouteNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>des_perf</td>
<td>0.54</td>
<td>17</td>
<td>56</td>
<td>54</td>
<td>42</td>
<td>74</td>
</tr>
<tr>
<td>edit_dist</td>
<td>1.00</td>
<td>25</td>
<td>36</td>
<td>38</td>
<td>28</td>
<td>64</td>
</tr>
<tr>
<td>fft</td>
<td>0.30</td>
<td>21</td>
<td>45</td>
<td>54</td>
<td>31</td>
<td>71</td>
</tr>
<tr>
<td>matrix_mult_a</td>
<td>0.21</td>
<td>13</td>
<td>30</td>
<td>34</td>
<td>12</td>
<td>49</td>
</tr>
<tr>
<td>matrix_mult_b</td>
<td>0.24</td>
<td>13</td>
<td>37</td>
<td>41</td>
<td>20</td>
<td>53</td>
</tr>
<tr>
<td>Average</td>
<td>0.46</td>
<td>18</td>
<td>41</td>
<td>44</td>
<td>27</td>
<td>62</td>
</tr>
</tbody>
</table>
DRC Hotspot Detection Evaluation

LR  Ground Truth  RouteNet
DRC Hotspot Detection Evaluation

• Variations of FCN
  • Infer seen: Training and inference on different placements of the same circuit
  • Less data: Trained on data from less designs
  • No short: Shortcut structure is removed
  • Less conv: Three convolutional layers are removed
  • No pool: Pooling layers are removed

<table>
<thead>
<tr>
<th>Circuit Name</th>
<th>FPR (%)</th>
<th>TPR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Infer seen</td>
</tr>
<tr>
<td>des_perf</td>
<td>0.54</td>
<td>77</td>
</tr>
<tr>
<td>edit_dist</td>
<td>1.00</td>
<td>68</td>
</tr>
<tr>
<td>fft</td>
<td>0.30</td>
<td>74</td>
</tr>
<tr>
<td>matrix_mult_a</td>
<td>0.21</td>
<td>51</td>
</tr>
<tr>
<td>matrix_mult_b</td>
<td>0.24</td>
<td>58</td>
</tr>
<tr>
<td>Average</td>
<td>0.46</td>
<td>66</td>
</tr>
</tbody>
</table>

Importance of large receptive region and global information.
DRC Hotspot Detection Evaluation

• Variations of baselines
  • 5×5: Use window size of 5×5 grid cells to capture neighboring features of each grid cell.
  • 9×9: 9 × 9 grid cells of window size.

<table>
<thead>
<tr>
<th>Circuit Name</th>
<th>FPR (%)</th>
<th>TPR (%)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>LR 5×5 LR</td>
<td>9×9 LR</td>
<td>SVM 5×5 LR</td>
<td>9×9 LR</td>
</tr>
<tr>
<td>des_perf</td>
<td>0.54</td>
<td>54</td>
<td>58</td>
<td>58</td>
<td>42</td>
<td>47</td>
</tr>
<tr>
<td>edit_dist</td>
<td>1.00</td>
<td>38</td>
<td>39</td>
<td>38</td>
<td>28</td>
<td>29</td>
</tr>
<tr>
<td>fft</td>
<td>0.30</td>
<td>54</td>
<td>56</td>
<td>54</td>
<td>31</td>
<td>41</td>
</tr>
<tr>
<td>matrix_mult_a</td>
<td>0.21</td>
<td>34</td>
<td>36</td>
<td>35</td>
<td>12</td>
<td>32</td>
</tr>
<tr>
<td>matrix_mult_b</td>
<td>0.24</td>
<td>41</td>
<td>44</td>
<td>42</td>
<td>20</td>
<td>39</td>
</tr>
<tr>
<td>Average</td>
<td>0.46</td>
<td>44</td>
<td>47</td>
<td>45</td>
<td>27</td>
<td>38</td>
</tr>
</tbody>
</table>

Large receptive region gives better results

But even larger window blurs local information

RouteNet is better choice
Conclusion

• We propose RouteNet:
  • Enables a global view for less homogeneous layout
  • Faster overall routability forecast at placement
  • More accurate hotspot detection at global routing
Thanks

zhiyao.xie@duke.edu
yiran.chen@duke.edu
haoxingr@nvidia.com
jianghu@tamu.edu