工業技術研究院
Industrial Technology Research Institute

# Fast and Accurate Image Recognition Using Deeply-Fused Branchy Networks

Mou-Yue Huang
Ching-Hao Lai
Sin-Horng Chen
CITC, ITRI
National Chiao Tung University

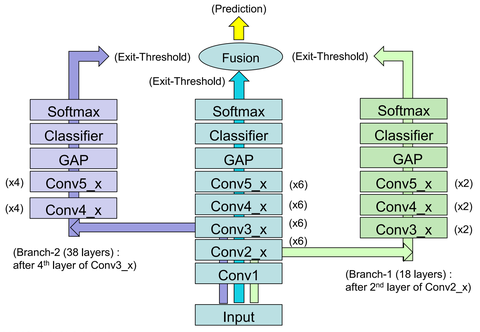


## *Deeply-Fused Branchy Network*

- Small-but-complete side branches
  - fewer layers, fewer output channels, faster inference
  - fully formed architectures resembling the main stream
  - make branches still achieve good enough accuracy
  - increase the ratio of exit samples at earlier branches
- Sequential decision making
  - early exit for easy-to-discriminate samples
  - save computation time
- Collaborative decision making
  - make probability fusion for hard-to-discriminate samples
  - improve accuracy




## Forward Inference



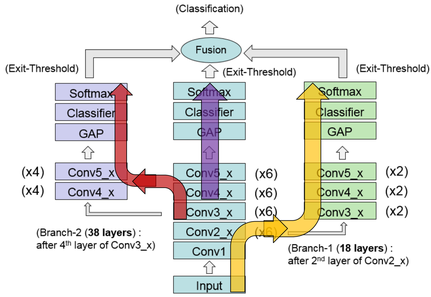

(DFB-WRN-N6k6-N2k2-N4k4)




## Inference Path

- Right stream ➔ Left stream ➔ Central main stream

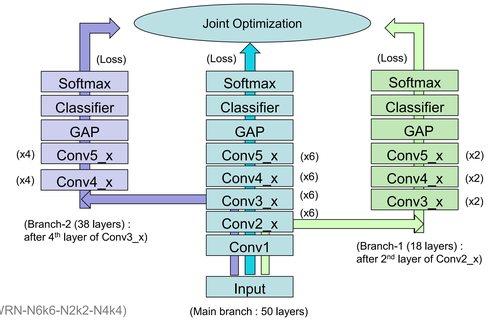
(Shorter) ---------- (Longer)







## Joint Optimization During Training




(DFB-WRN-N6k6-N2k2-N4k4)




## Building Blocks of DFB-Net

| Group Name | Output Size | Block Type |
|---|---|---|
| Conv1 | 56 x 56 | [3 x 3, 16] |
| Conv2_x | 56 x 56 | $\begin{bmatrix} 3 \times 3, & 16 \times k \\ 3 \times 3, & 16 \times k \end{bmatrix} \times N$ |
| Conv3_x | 28 x 28 | $\begin{bmatrix} 3 \times 3, & 32 \times k \\ 3 \times 3, & 32 \times k \end{bmatrix} \times N$ |
| Conv4_x | 14 x 14 | $\begin{bmatrix} 3 \times 3, & 64 \times k \\ 3 \times 3, & 64 \times k \end{bmatrix} \times N$ |
| Conv5_x | 7 x 7 | $\begin{bmatrix} 3 \times 3, & 128 \times k \\ 3 \times 3, & 128 \times k \end{bmatrix} \times N$ |
| Global-Ave-Pool | 1 x 1 | [7 x 7] |




**Algorithm 1:** DFB-Net Forward Inference

**Input:** A test image $\boldsymbol{x}$, exit thresholds $\{p_s\}$
**Output:** The predicted label of test image $\boldsymbol{x}$

**procedure** DFB-Net($\boldsymbol{x}$, $\{p_s\}$)
  Initialize $\bar{\boldsymbol{y}} = 0$
  **for** $s = 1, \ldots, M$ **do**
    $\boldsymbol{z} = f(\boldsymbol{x}; W_s)$
    $\boldsymbol{y} = \text{softmax}(\boldsymbol{z})$
    **if** $\max\{\boldsymbol{y}\} > p_s$ **then**
      **return** $argmax\{\boldsymbol{y}\}$
    **else**
      $\bar{\boldsymbol{y}} = \bar{\boldsymbol{y}} + \boldsymbol{y}$
  $\bar{\boldsymbol{y}} = \bar{\boldsymbol{y}} / M$
  **return** $argmax\{\bar{\boldsymbol{y}}\}$




## Time-Accuracy Trade-off

- For time critical application, simply set the exit threshold to be 0 at the 1st earliest branch
- For accuracy critical mission, set the exit thresholds of earlier branches to be higher (e.g., 0.99)
- Exit thresholds provide the flexibility of controlling the trade-off between computation time and accuracy

## Approach to Training Side Branches

- Firstly, train the main stream net from scratch
- Secondly, load the already-trained main stream model to fine-tune its branchy sub-nets




## Experiment Setup

- Basically CIFAR-10 and CIFAR-100 use the same DFB-Net architecture
- Apply Dropout or not
  - CIFAR-10 ➔ No
  - CIFAR-100 ➔ Yes
- Down-sampling method
  - CIFAR-10 ➔ Use Conv_1x1 with Stride 2
  - CIFAR-100 ➔ Use 2x2 Ave-Pool
- Apply scale and aspect ratio data augmentation
- Use GTX-1080, CUDA 8.0, cuDNN 5.1 for inference




**Performance Results : CIFAR-10**

| Network Topology | Exit Thresholds (Exit-1, Exit-2, Exit-3) | Error (%) | Time (ms) | Gain (x) | Exit Ratio (%) (Exit-1, Exit-2, Exit-3, Fused) | Error (%) within Each Branch (Exit-1, Exit-2, Exit-3, Fused) |
|---|---|---|---|---|---|---|
| (Baseline) WRN-50-N6-k6 | N/A | 3.23 | 29.67 | 1.00 | N/A | N/A |
| DFB-Net : (Exit-1) Branch-1, 18 layers (Exit-2) Branch-2, 38 layers (Exit-3) Baseline, 50 layers | 0.900, 0.900, 0.00 | 3.72 | 7.39 | 4.01 | 90.48, 5.98, 3.54 | 1.90, 15.72, 29.94 |
| | 0.900, 0.900, 0.75 | 3.63 | 7.43 | 3.99 | 90.48, 5.98, 2.83, 0.71 | 1.90, 15.72, 23.32, 43.66 |
| | 0.950, 0.950, 0.00 | 3.54 | 8.21 | 3.61 | 87.50, 7.05, 5.45 | 1.37, 11.21, 28.44 |
| | 0.950, 0.950, 0.75 | 3.39 | 8.22 | 3.61 | 87.50, 7.05, 4.50, 0.95 | 1.37, 11.21, 22.67, 40.00 |
| | 0.975, 0.975, 0.00 | 3.46 | 9.09 | 3.26 | 84.33, 8.27, 7.40 | 1.01, 7.86, 26.49 |
| | 0.975, 0.975, 0.75 | 3.29 | 9.14 | 3.25 | 84.33, 8.27, 6.30, 1.10 | 1.01, 7.86, 21.59, 39.09 |
| | 0.990, 0.975, 0.00 | 3.36 | 9.85 | 3.01 | 80.03, 11.53, 8.44 | 0.65, 5.98, 25.48 |
| | 0.990, 0.975, 0.75 | **3.15** | 9.89 | **3.00** | 80.03, 11.53, 7.22, 1.22 | 0.65, 5.98, 20.50, 37.70 |
| | 0.990, 0.990, 0.00 | 3.29 | 10.35 | 2.87 | 80.03, 9.48, 10.49 | 0.65, 4.11, 22.69 |
| | 0.990, 0.990, 0.75 | **3.07** | 10.41 | **2.85** | 80.03, 9.48, 9.19, 1.30 | 0.65, 4.11, 18.06, 38.46 |

| Method | Depth | Params | C10 | C10+ | C100 | C100+ |
|---|---|---|---|---|---|---|
| Wide ResNet [41] | 16 | 11.0M | - | 4.81 | - | 22.07 |
| | 28 | 36.5M | - | 4.17 | - | 20.50 |
| with Dropout | 16 | 2.7M | - | - | - | - |
| ResNet (pre-activation) [12] | 164 | 1.7M | 11.26* | 5.46 | 35.58* | 24.33 |
| | 1001 | 10.2M | 10.56* | 4.62 | 33.47* | 22.71 |
| DenseNet ($k = 12$) | 40 | 1.0M | **7.00** | 5.24 | **27.55** | 24.42 |
| DenseNet ($k = 12$) | 100 | 7.0M | **5.77** | **4.10** | **23.79** | **20.20** |
| DenseNet ($k = 24$) | 100 | 27.2M | **5.83** | **3.74** | **23.42** | **19.25** |
| DenseNet-BC ($k = 12$) | 100 | 0.8M | **5.92** | 4.51 | **24.15** | 22.27 |
| DenseNet-BC ($k = 24$) | 250 | 15.3M | **5.19** | **3.62** | **19.64** | **17.60** |
| DenseNet-BC ($k = 40$) | 190 | 25.6M | - | **3.46** | - | **17.18** |

CIFAR-10 Experiment Results

[CVPR, 2017] Densely Connected Convolutional Networks




**Performance Results : CIFAR-100**

| Network Topology | Exit Thresholds (Exit-1, Exit-2, Exit-3) | Error (%) | Time (ms) | Gain (x) | Exit Ratio (%) (Exit-1, Exit-2, Exit-3, Fused) | Error (%) within Each Branch (Exit-1, Exit-2, Exit-3, Fused) |
|---|---|---|---|---|---|---|
| (Baseline) WRN-50-N6-k6 | N/A | 17.74 | 29.39 | 1.00 | N/A | N/A |
| DFB-Net : (Exit-1) Branch-1, 18 layers (Exit-2) Branch-2, 38 layers (Exit-3) Baseline, 50 layers | 0.75, 0.75, 0.00 | 18.06 | 10.01 | 2.94 | 78.73, 11.47, 9.80 | 10.91, 34.70, 56.02 |
| | 0.75, 0.75, 0.75 | 17.89 | 10.02 | 2.93 | 78.73, 11.47, 4.38, 5.42 | 10.91, 34.70, 38.58, 66.97 |
| | 0.80, 0.75, 0.00 | 17.78 | 10.62 | 2.77 | 75.83, 13.43, 10.74 | 9.75, 33.43, 54.93 |
| | 0.80, 0.75, 0.75 | **17.55** | 10.67 | **2.75** | 75.83, 13.43, 4.93, 5.81 | 9.75, 33.43, 37.93, 65.40 |
| | 0.85, 0.80, 0.00 | 17.34 | 11.51 | 2.55 | 72.62, 14.32, 13.06 | 8.39, 29.19, 54.13 |
| | 0.85, 0.80, 0.75 | 17.09 | 11.52 | 2.55 | 72.62, 14.32, 6.18, 6.88 | 8.39, 29.19, 37.70, 65.26 |
| | 0.90, 0.90, 0.00 | 16.94 | 13.04 | 2.25 | 68.64, 13.50, 17.86 | 6.98, 23.11, 50.56 |
| | 0.90, 0.90, 0.75 | 16.64 | 13.06 | 2.25 | 68.64, 13.50, 9.25, 8.61 | 6.98, 23.11, 35.35, 63.41 |
| | 0.95, 0.85, 0.00 | 16.64 | 13.77 | 2.13 | 62.61, 19.73, 17.66 | 4.87, 22.76, 51.53 |
| | 0.95, 0.85, 0.75 | 16.42 | 13.81 | 2.13 | 62.61, 19.73, 9.06, 8.60 | 4.87, 22.76, 36.53, 64.77 |
| | 0.99, 0.99, 0.00 | 16.60 | 18.81 | 1.56 | 50.79, 14.68, 34.53 | 2.30, 8.92, 40.89 |
| | 0.99, 0.99, 0.75 | **16.01** | 18.83 | **1.56** | 50.79, 14.68, 21.91, 12.62 | 2.30, 8.92, 27.89, 58.80 |

| Method | Depth | Params | C10 | C10+ | C100 | C100+ |
|---|---|---|---|---|---|---|
| Wide ResNet [41] | 16 | 11.0M | - | 4.81 | - | 22.07 |
| | 28 | 36.5M | - | 4.17 | - | 20.50 |
| with Dropout | 16 | 2.7M | - | - | - | - |
| ResNet (pre-activation) [12] | 164 | 1.7M | 11.26* | 5.46 | 35.58* | 24.33 |
| | 1001 | 10.2M | 10.56* | 4.62 | 33.47* | 22.71 |
| DenseNet ($k = 12$) | 40 | 1.0M | **7.00** | 5.24 | **27.55** | 24.42 |
| DenseNet ($k = 12$) | 100 | 7.0M | **5.77** | **4.10** | **23.79** | **20.20** |
| DenseNet ($k = 24$) | 100 | 27.2M | **5.83** | **3.74** | **23.42** | **19.25** |
| DenseNet-BC ($k = 12$) | 100 | 0.8M | **5.92** | 4.51 | **24.15** | 22.27 |
| DenseNet-BC ($k = 24$) | 250 | 15.3M | **5.19** | **3.62** | **19.64** | **17.60** |
| DenseNet-BC ($k = 40$) | 190 | 25.6M | - | **3.46** | - | **17.18** |

CIFAR-100 Experiment Results

[CVPR, 2017] Densely Connected Convolutional Networks




## Summary

- Our CIFAR-10 baseline model achieves state-of-the-art result with **3.23%** error rate
- On CIFAR-10, our branchy network with fusion
  - achieve state-of-the-art result with **3.07%** error rate
  - got **3.0x** speedup while better than baseline model
- On CIFAR-100, our branchy networks with fusion
  - achieve state-of-the-art result with **16.01%** error rate
  - got **2.75x** speedup while better than baseline model