Supplementary Material for "DualNet: Learn Complementary Features for Image Recognition"

Saihui Hou, Xu Liu and Zilei Wang
Department of Automation, University of Science and Technology of China
{saihui, liuxu91}@mail.ustc.edu.cn, zlwang@ustc.edu.cn

1. CIFAR-100 Augmentation

The parameters for data augmentation when training ResNet-20 [6] on CIFAR100 [7] are listed as follows. And the first three parameters are for changing contrast and brightness of input images (please see [1] for the algorithm of changing image contrast and brightness).

- $min\_contrast = 0.8$: minimum contrast multiplier (min α).
- $max\_contrast = 1.2$: maximum contrast multiplier (max α).
- $max\_brightness\_shift = 5$: maximum brightness shift in positive and negative directions (β).
- $max\_color\_shift = 5$: maximum color shift along RGB axes.
- $apply\_probability = 0.5$: how often each transformation is applied.
- zero-padding with 2 pixels for each side and crop with 32×32 for training.

2. DualNet From ResNet-32&ResNet-56

Besides ResNet-20, we further evaluate DualNet based on the deeper ResNet [6], *e.g.*, with 32 layers and 56 layers (denoted as ResNet-32&ResNet-56, referring to the third-party implementation available at [2]). ResNet-32&ResNet-56, as well as the corresponding DualNet (denoted as DNR32&DNR56), are also trained on the augmented CIFAR-100 and the experimental results are shown in Table 1. The performance comparison demonstrates that DualNet performs well with ResNet-32 and ResNet-56. For DNR56, although the *joint finetuning* does not helps a lot, the performance (75.57%) is still 2.76% higher than the base model (72.81%).

Table 1. **The top-1 accuracy on the augmented CIFAR-100 achieved by the standard deep model (ResNet-32&ResNet-56) and the corresponding DualNet (DNR32&DNR56).** The first row shows the results of ResNet-32&ResNet-56 and the rest are all achieved by DNR32&DNR56. After the *iterative training*, we respectively evaluate the performance of the *Fused Classifier* and the weighted average of three classifiers, while the latter one can help validate the necessity of the *joint finetuning*.

Training / Model	DNR32	DNR56
standard deep model (ResNet-32&ResNet-56)	69.72%	72.81%
iterative training (*Fused Classifier*)	73.06%	75.24%
iterative training (classifier average)	73.31%	75.53%
joint finetuning (classifier average)	**73.51%**	**75.57%**

3. More Experimental Analyses

In Section 4.4 of manuscript, we have presented some experimental analyses about the design of DualNet. Here more experimental analyses are provided to further validate its robustness and superiority.

3.1. Comparison to global finetuning

In the manuscript we have tried to finetune the DualNet globally with the auxiliary classifiers removed and the results are given in Section 4.4. Here we further try to finetune the whole DualNet globally regardless of training time and memory cost. NIN [9] is chosen as the base network because of its small size, and then the whole DNI is globally finetuned from scratch on CIFAR100 without data augmentation. Finally we get 69.36% on CIFAR100 by averaging the output of three classifiers, which is still lower than that achieved by taking the proposed procedure (69.76%), although much more computation cost is introduced.

3.2. Comparison to model ensemble

In addition, we compare our DualNet to the model ensemble technique [8], *i.e.*, an ensemble of networks trained independently and then averaged for their final prediction, which is also taken for comparison with HD-CNN in [11].

Table 2. **Performance comparison with model ensemble technique on CIFAR100 (without augmentation) and Stanford Dogs.** All results are reported in top-1 mean accuracy.

Model \ Dataset	CIFAR100	Model \ Dataset	Stanford Dogs
NIN(Single)	66.91%	VGG(Single)	74.11%
NIN(Ensemble)	**70.18%**	VGG(Ensemble)	75.77%
DNI(Single)	69.76%	DNV(Single)	**77.56%**

In practice, we train two networks of the same architecture independently and then average their output for prediction to compare with DualNet which consists of two subnetworks. Due to the randomness in training, the resulting parameters of independent networks are different, and model ensemble can help improve the accuracy for recognition. The performance comparison is shown in Table 2 and the results are reported on CIFAR100 and Stanford Dogs. On CIFAR100, DNI is a little inferior but approximate to the ensemble of NIN. On Stanford Dogs, DNV achieves higher accuracy than the ensemble of VGGNet. As stated in [11], the model ensemble technique requires training and evaluation of independent models, while our DualNet as well as HD-CNN is end-to-end in a unified framework, which is orthogonal to the model ensemble technique.

3.3. Fusion at various layers

In this subsection, we take the experiments of fusion at various layers and the results are reported on VGGNet and Stanford Dogs. The output of Fused Classifier after the *iterative training* is taken for the evaluation. Then we get the following results when fusing at different layers: *pool5*-77.03%, *pool4*-74.72%, *pool3*-73.94%, *pool2*-73.68%, *pool1*-73.85%. The performance comparison shows that fusing at *pool5* makes sense in this case.

3.4. Fusion weights analysis

In our experiments the fusion weights of *S1 Classifier* and *S2 Classifier* are empirically set to 0.3. Here we try other fusion weights especially when the weights of *S1 Classifier* and *S2 Classifier* are larger than that of *Fused Classifier*. The experiments are taken on DNV and Stanford Dogs and the output of Fused Classifier after the *iterative training* is taken for the evaluation. Then we get the following results when using different loss weights: 0.3-77.03%, 0.5-76.39%, 1.0-75.45%, 2.0-74.94%. The results indicate that the fusion weights used in the manuscript are reasonable.

3.5. Evaluation of N extractors

In this work, we coordinate two extractors to learn complementary features and it would be interesting to extend to N (more than two) extractors. Here we take a simple experiment on NIN and CIFAR100 (without data augmentation) and explore the cooperation of N extractors. Specifically, the N extractors are added gradually and the N-stream features are summed into the Fused Classifier for recognition. In the training, the parameters of the newly added N-th extractor are updated with the rest N-1 fixed, while an auxiliary classifier is appended to the N-th extractor to make the features produced by it discriminative alone. To enable a fair comparison, the output of Fused Classifier is taken for the evaluation through the experiments. Then we get the following results when using different number of extractors: One-66.91% (baseline), Two-69.01%, Three-69.87%, Four-69.74%. The results show that the performance of four extractors is slightly worse than that of three extractors, indicating that more extractors are not always beneficial. More advanced strategy is needed to coordinate the N extractors, which is considered for our future work.

4. Datasets to Evaluate DualNet

The recent work, DDN [10], which is directly related to ours, mainly conducts experiments on CIFAR100. While in our work, besides CIFAR100, DualNet is also evaluated on the widely-used Standford Dogs and UEC FOOD-100. For the large-scale ImageNet, training from scratch is very time-consuming and requires multiple GPUs in some cases, *e.g.*, the training of ResNet on ImageNet requires parallel computing of up to 8 GPUs [3]. In contrast, all the networks in the manuscript can be trained/tested on a Tesla K40 GPU. Here we choose NIN-ImageNet[1] which is relatively fast for training as the base network, and evaluate the effectiveness of DualNet on the ILSVRC-2012 ImageNet dataset. NIN-ImageNet greatly reduces the number of parameters compared to AlexNet [8] but reports similar accuracy on ImageNet. The experimental results are shown in Table 3 and reported on the validation set. It can be seen that, DualNet from NIN-ImageNet achieves 1.29% higher top-1 accuracy than the base network on the large-scale ImageNet.

In summary, the proposed DualNet performs well with different DCNN architectures and various datasets, and reports performance superior to the baselines in all cases. The improvement in certain cases may be not that significant but convincing, indicating that DualNet can really help acquire more discriminative image representation by coordinating two DCNNs to learn complementary features, which is the key idea of this work.

5. Visualization of Network Architectures

In this section, we provide the network architectures that are evaluated in the manuscript, including CaffeNet, VG-

[1]The architectures of NIN-ImageNet (NIN for ImageNet here) and NIN-CIFAR100 (NIN for CIFAR100 used in the manuscript) are a little different, both of which are available in the Caffe Model Zoo [4]. Please refer to [4] for more details.

Table 3. **Performance comparison on the ILSVRC-2012 ImageNet between NIN-ImageNet and the corresponding DualNet.** All results are reported in top-1 mean accuracy.

Model \ Dataset	ILSVRC-2012 ImageNet
NIN-ImageNet	59.15%
DualNet from NIN-ImageNet	**60.44%**

GNet, NIN (for CIFAR100), ResNet-20 and their corresponding DualNet. All the network structures are visualized using the tools provided by [5]. Best viewed electronically.

5.1. CaffeNet (Figure 1)

5.2. DualNet From CaffeNet (Figure 2)

5.3. VGGNet (Figure 3)

5.4. DualNet From VGGNet (Figure 4)

5.5. NIN (Figure 5)

5.6. DualNet From NIN (Figure 6)

5.7. ResNet-20 (Figure 7)

5.8. DualNet From ResNet-20 (Figure 8)

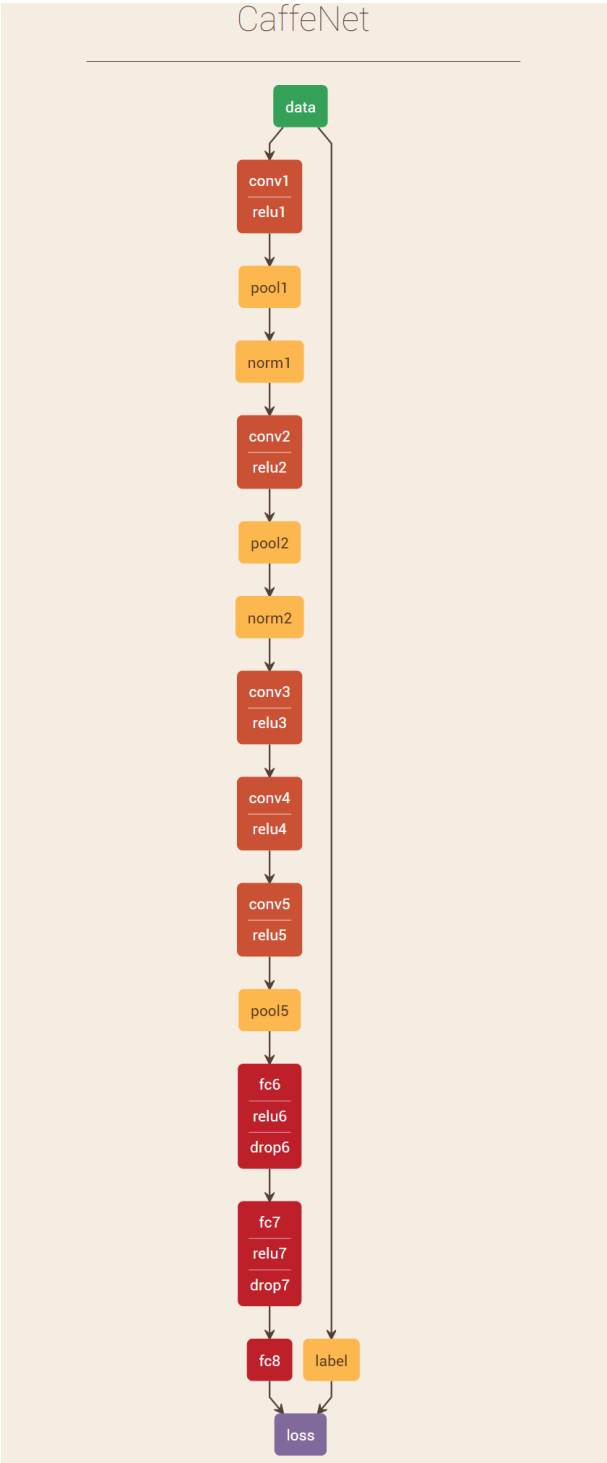

Figure 1. **The architecture of CaffeNet**.

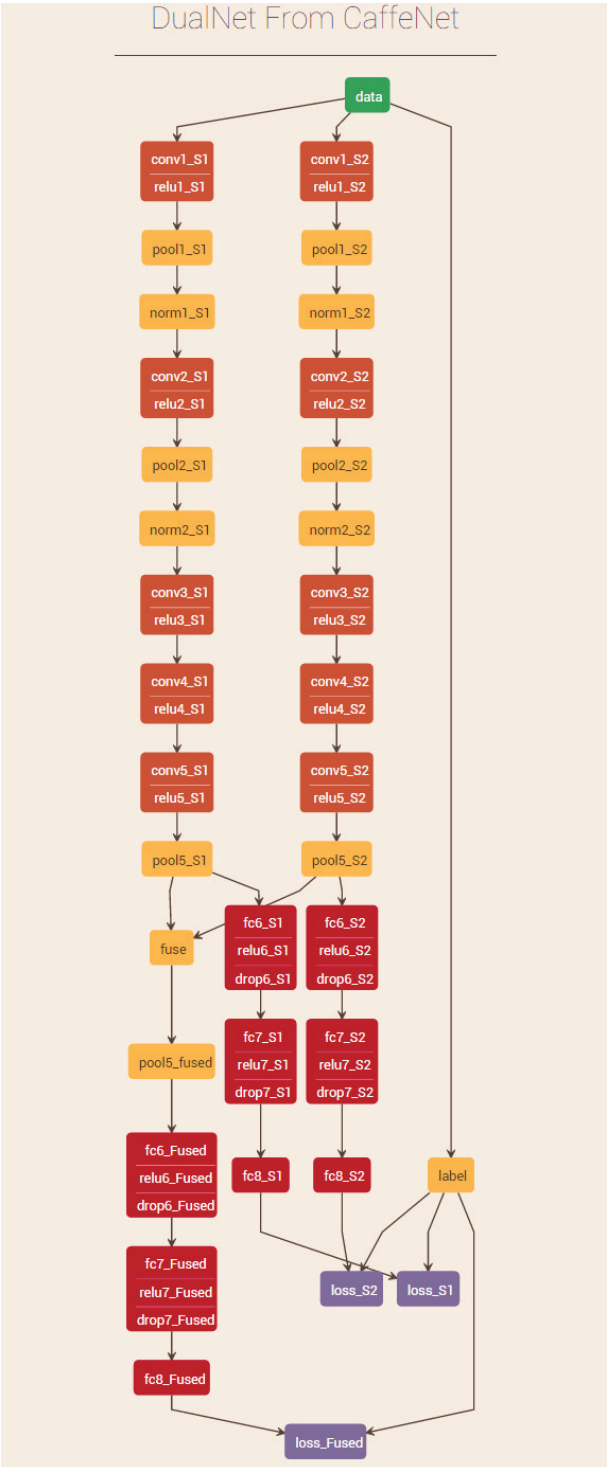

Figure 2. **The architecture of DualNet From CaffeNet**.

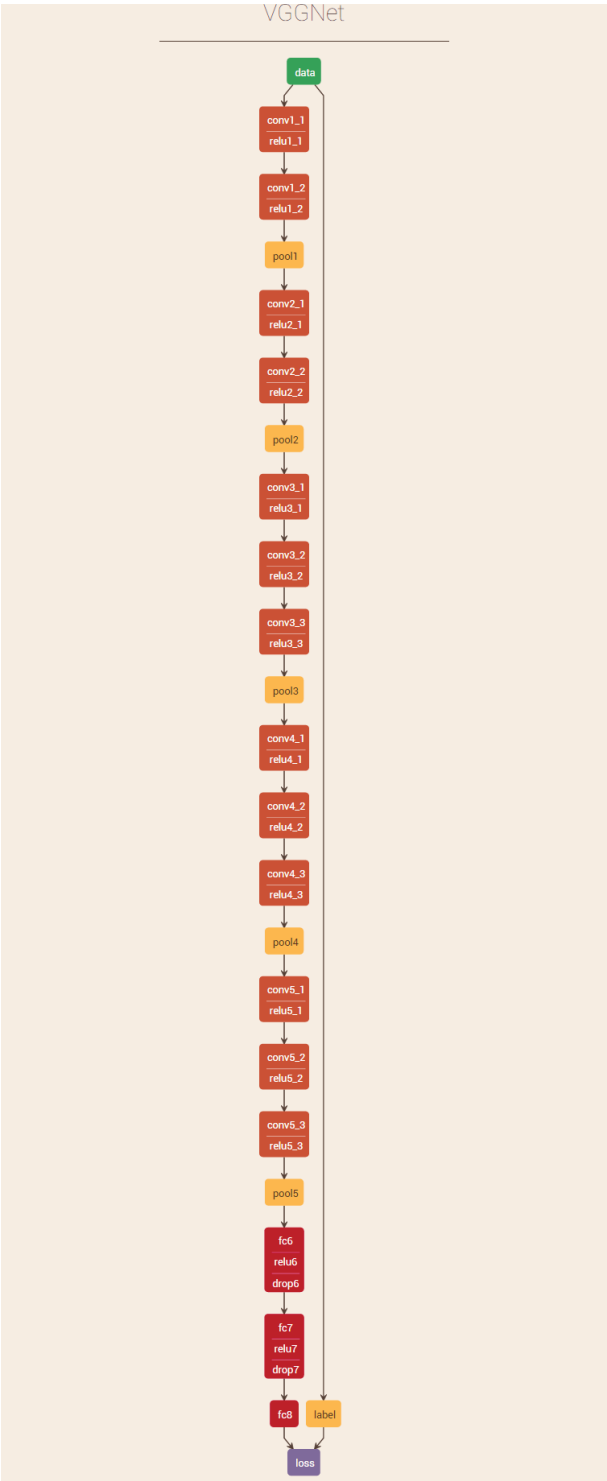

Figure 3. **The architecture of VGGNet**.

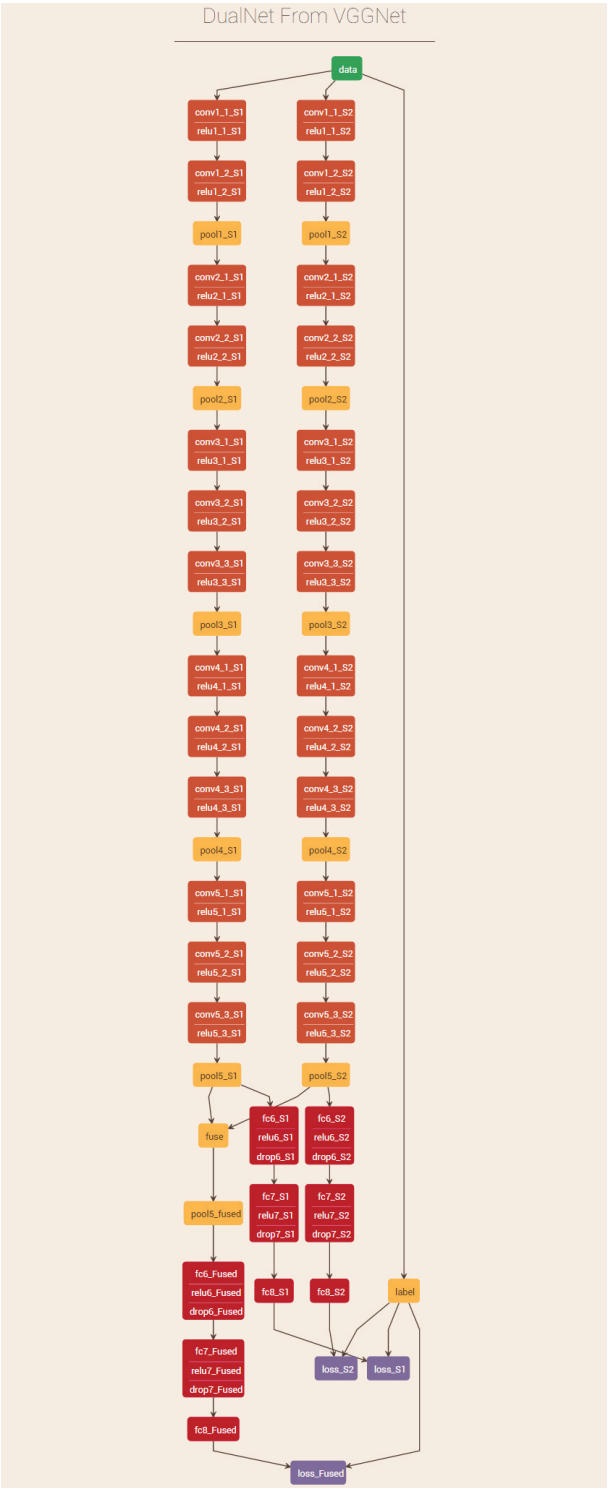

Figure 4. **The architecture of DualNet From VGGNet**.

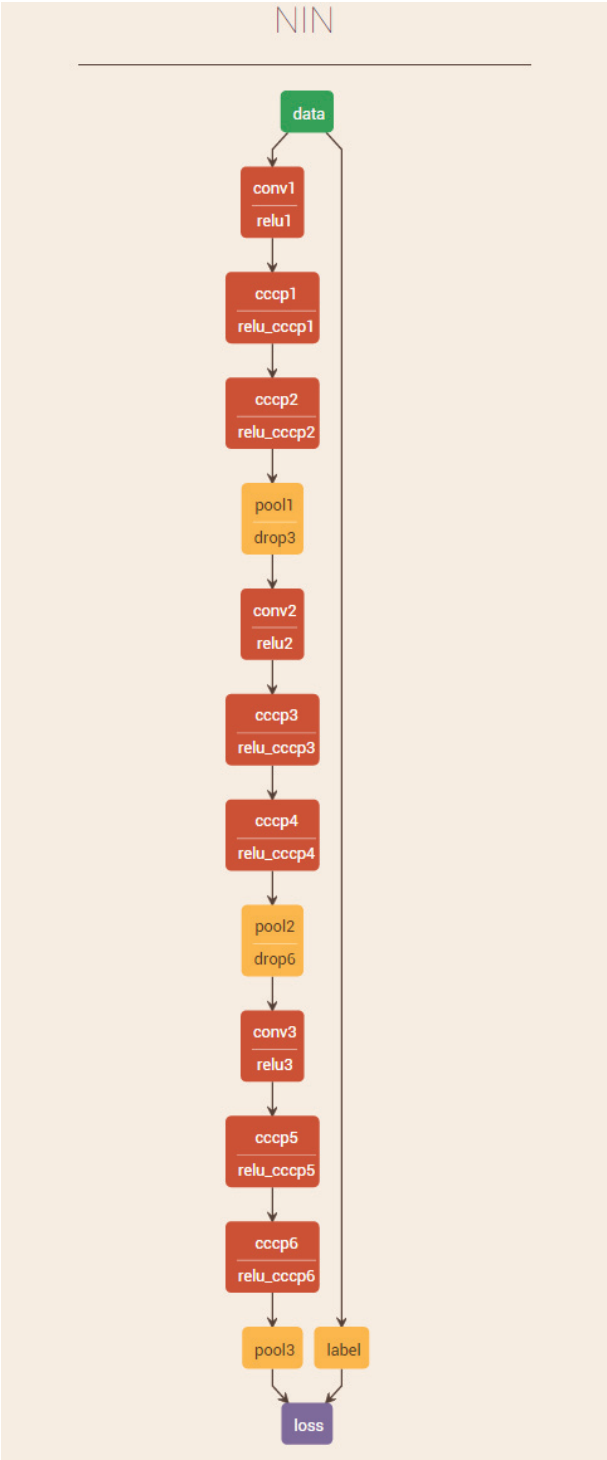

Figure 5. **The architecture of NIN**.

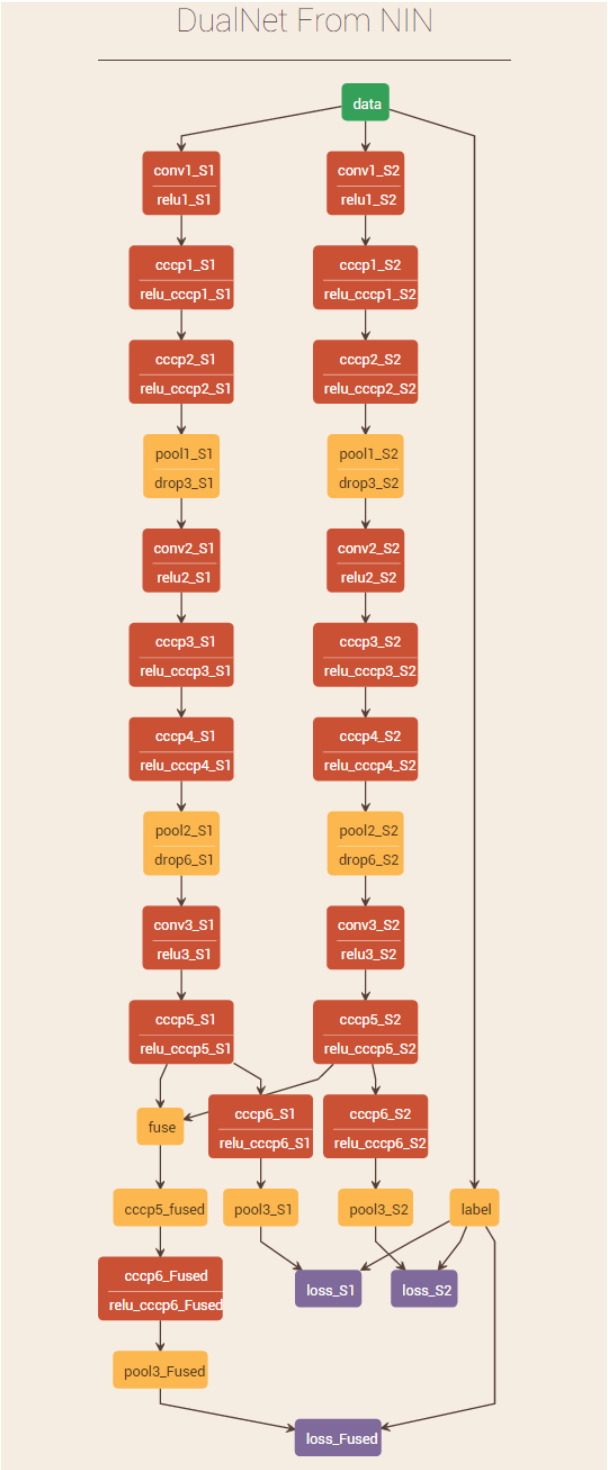

Figure 6. **The architecture of DualNet From NIN**.

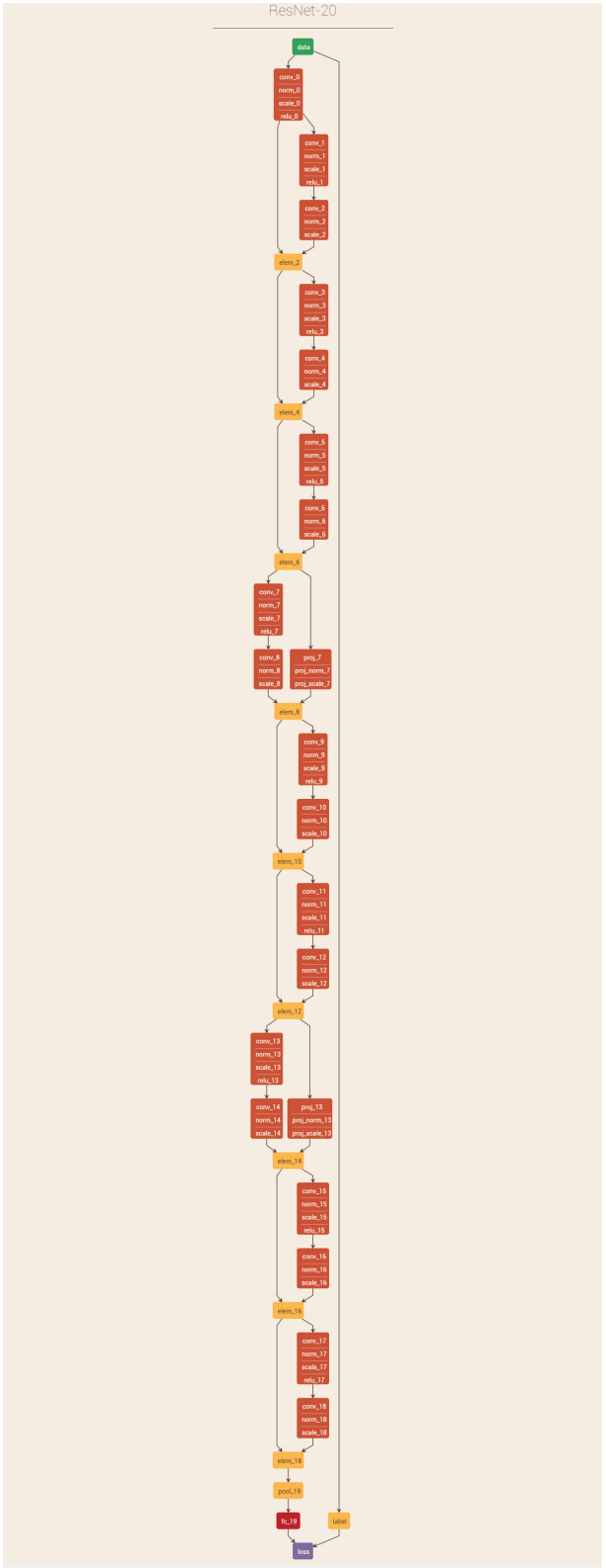

Figure 7. **The architecture of ResNet-20**.

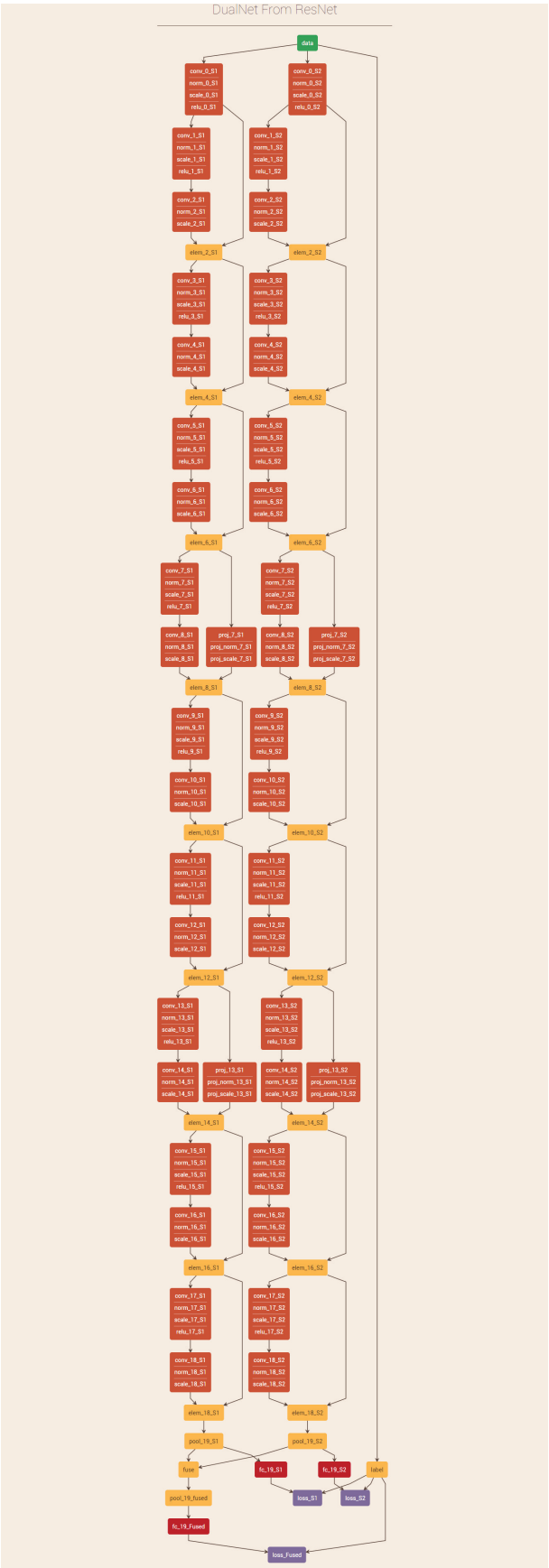

Figure 8. **The architecture of DualNet From ResNet-20**.

References

[1] http://docs.opencv.org/2.4/doc/tutorials/core/basic_linear_transform/basic_linear_transform.html. 1

[2] https://github.com/twtygqyy/resnet-cifar10. 1

[3] https://github.com/KaimingHe/deep-residual-networks. 2

[4] https://github.com/BVLC/caffe/wiki/Model-Zoo. 2

[5] http://ethereon.github.io/netscope/quickstart.html. 3

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[7] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009. 1

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 2

[9] M. Lin, Q. Chen, and S. Yan. Network in network. In *ICLR*, 2014. 1

[10] V. N. Murthy, V. Singh, T. Chen, R. Manmatha, and D. Comaniciu. Deep decision network for multi-class image classification. In *CVPR*, 2016. 2

[11] Z. Yan, H. Zhang, R. Piramuthu, V. Jagadeesh, D. DeCoste, W. Di, and Y. Yu. Hd-cnn: Hierarchical deep convolutional neural networks for large scale visual recognition. In *ICCV*, 2015. 1, 2