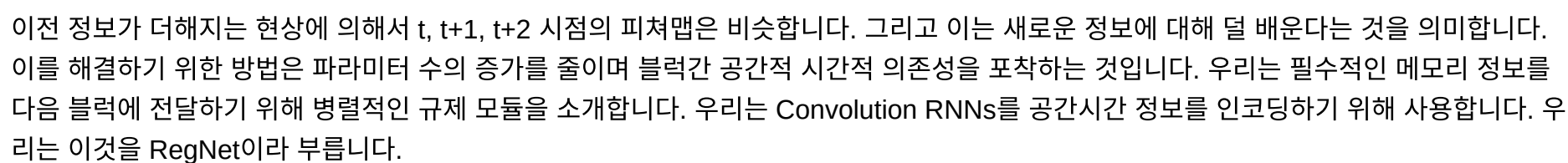
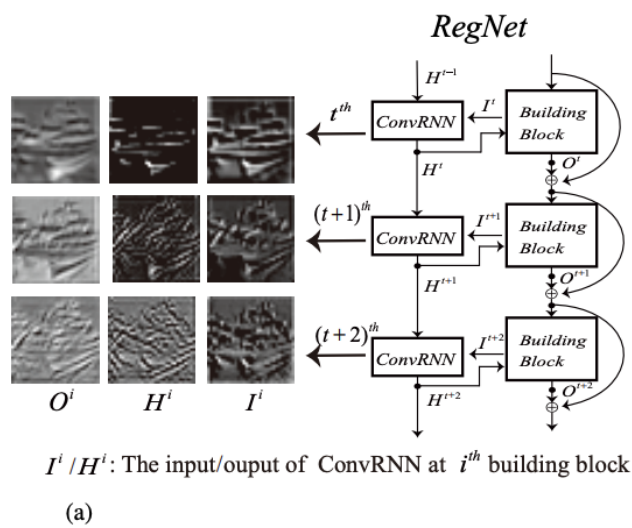




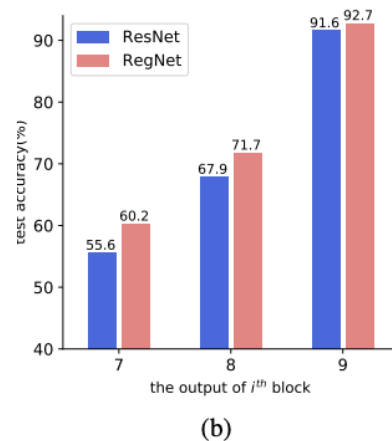
Resnet과 그 변형된 다른 모델들은 컴퓨터 비전 테스트에서 획기적인 성공을 거두었습니다. 성공하였지만 building block을 통한 gradient flow는 기존 정보를 더하는 연산으로 인해 새롭게 나타난 정보에 대한 학습이 부족했습니다. 이러한 문제를 해결하기 위해 우리는 상호보완적 피쳐를 추출하는 메모리 메카니즘의 규제 모듈을 제시합니다. 이 규제 모듈은 conv RNNs로 구성되어 있으며 공간적-시간적 정보를 잘 추출해 냅니다. 우리는 이러한 네트워크를 RegNet이라 칭합니다. 규제 모듈은 기존 모든 ResNet 구조에 쉽게 적용할 수 있습니다. 우리는 우리가 제안한 방식을 3가지 classification dataset에서 우수함을 증명합니다.

CNN네트워크는 많은 CV 테스트에서 획기적인 성과를 이루어냈습니다. 특히 ResNet 계열은 큰 성과를 이루어냈습니다. ResNet의 성공 이유 중 가장 큰 것은 더 깊은 학습을 가능케 하는 shortcut connection mechanism(residual-block)때문입니다. 그러나 이러한 메커니즘은 각각이 블럭이 residual output에 집중하게 만들고 전개된 정보는 무시됩니다. 그리고 재사용 가능한 이전 블럭에서의 정보는 이후 블럭에서 점차 없어지는 경향이 있습니다. 이를 설명하기 위해 우리는 ResNet featuremap에 대해 시각화 합니다.





RegNet은 병렬적인 규제 모듈을 사용해 ConvRNN과 Building Block에서 나오는 정보를 이용하여 다음 블록에 전달합니다. H는 공간적 시간적 정보를 담을 수 있게 되며 O는 기존 ResNet의 아웃풋보다 풍부한 정보를 담게 됩니다.



이러한 모듈을 추가했을 때 성능을 확인할 경우 실제로 RegNet은 ResNet보다 더 뛰어난 것을 확인할 수 있습니다. 게다가 RegNet은 병렬적인 모듈이기 때문에 대부분의 ResNet 변형 모델에도 쉽게 적용할 수 있습니다.

평가를 위해서 우리는 3가지 CIFAR-10, CIFAR-100, ImageNet 데이터셋을 이용합니다. 그리고 우리의 모델이 모든 데이터 셋에서 ResNet보다 성능이 우월함을 보입니다.

## RELATED WORK

딥러닝 네트워크는 머신러닝 분야에서 획기적인 돌파구가 되었습니다. 그러나, 깊은 학습 네트워크는 어려운 문제였습니다. Shortcut connection은 이러한 문제를 해결하였고, 정보는 감소 없이 다음 레이어로 전이될 수 있었습니다.

ResNet의 강력한 수용성으로 인해 많은 테스트에서 좋은 성능을 얻었습니다. 예를 들어 디텍션에서 ResNet은 feature extraction으로 사용되었습니다. 이러한 성공에도 불구하고 ResNet은 여전히 깊이 문제를 겪습니다. DenseNet의 경우 인풋 피쳐와 아웃풋 피쳐를 연결하여 사용합니다. 그러나 이는 많은 연산 비용을 요구합니다. ResNet에서 중간 피쳐를 활용하는 식의 많은 발전이 있었지만 지금까지 공간 시간적 특징을 완전히 활용한 경우는 없었습니다.

그러나, Conv RNN, Conv LSTM, Conv GRU는 많은 테스트에서 공간 시간 정보를 해석하기 위해 많이 사용되었으며 이미지 간 연관성 정보를 해석하는데 사용되었습니다. 그리고 우리는 Conv RNN을 이용해 공간 시간 정보를 추출하는 규제 모듈로 사용합니다.

## MODEL

먼저 우리는 ResNet의 배경과 ConvRNNs에서 진보된 ConvLSTM과 ConvGRU에 대해 살핀 후 RegNet을 소개합니다.

### ResNet

네트워크가 깊어짐에 따라 성능 저하의 문제는 발생했습니다. 이러한 문제는 ResNet의 Residual Block에 의해 해결됩니다. Residual Block은 원본 정보를 layer이후 출력물과 합쳐줌으로써 정보량의 감소를 줄여서 깊은 네트워크를 가능케 합니다. 이는 기울기 소실 문제를 해결합니다.

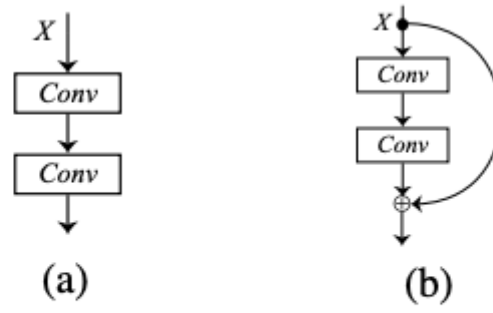


Fig. 2. 2(a) shows the original underlying mapping while 2(b) shows the residual mapping in ResNet [5].

### ConvRNN and its Variants

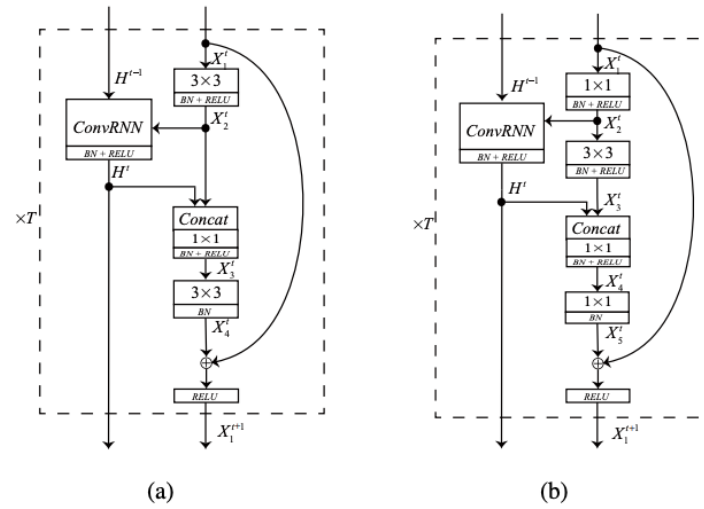
RNN, LSTM 그리고 GRU는 시퀀스 처리에서 굉장한 성공을 했습니다. 공간-시간 문제를 해결하기 위해 우리는 ConvRNN, ConvLSTM, ConvGRU를 이용합니다. 이는 RNN의 fully-connected layer를 convolution layer로 교체한 것입니다.

$$\mathbf{H}^t = \tanh(2^N \mathbf{W}_h^N * [\mathbf{X}^t, \mathbf{H}^{t-1}] + \mathbf{b}_h), \quad (1)$$

RNN에 들어오는 인풋은 각각의 N개의 채널을 가진 X와 H 총 두개이다. 이를 연산하기 위해 concat하여 2N의 피쳐 맵을 생성한다. 그리고 2N개의 채널을 가진 인풋은 N개의 그룹으로 나눈다. 각 그룹은 두개의 채널을 가지게 되고 여기에 1\*1conv를 사용해 이전 정보와 현재 정보를 합치는 1개의 채널을 생성한다. 이렇게 생성된 N개 그룹의 각각의 1개의 채널은 3\*3 conv를 통해 지역적 정보를 얻는다. 이렇게 3D conv를 사용하지 않고 각각의 conv로 나누는 것은 연산량 감소를 얻을 수 있다. 이러한 방식은 ConvGRU, ConvLSTM에도 비슷하게 사용된다.

### RNN-Regulated ResNet

평가에 앞서 우리는 두가지 방식의 ResNet building blocks를 제안한다. bottleneck block의 유무에 대한 것이다.



Bottleneck regnet은 bottleneck resnet으로 부터 참조하였다. 이를 통해 연산량 감소 효과를 얻는다.

## Experiments

### Results on CIFAR

**TABLE III**  
CLASSIFICATION ERROR RATES ON THE CIFAR-10/100. BEST RESULTS ARE MARKED IN BOLD.

model	C10	C100
ResNet-20 [5]	8.38	31.72
RegNet-20(ConvRNN)	7.60	30.03
RegNet-20(ConvGRU)	7.42	<b>29.69</b>
RegNet-20(ConvLSTM)	<b>7.28</b>	29.81
SE-ResNet-20	8.02	31.14
SE-RegNet-20(ConvRNN)	7.55	29.63
SE-RegNet-20(ConvGRU)	7.25	29.08
SE-RegNet-20(ConvLSTM)	<b>6.98</b>	<b>29.02</b>

SE-RegNet은 RegNet에서 레이어가 좀 더 추가되어진 모델. 기존 ResNet을 사용할 때 보다 규제 모듈을 추가했을 때 성능이 향상된 것을 확인할 수 있다.

## Parameters Analysis

TABLE IV  
TEST ERROR RATES ON CIFAR-10/100. WE USE CONVGRU AND CONVLSTM AS REGULATORS OF RESNET. WE LIST THE INCREASE OF PARAMETER THE ARCHITECTURES AT THE RIGHT CORNER OF THE ERROR RATES.

layer	C-10			C-100		
	ResNet	+ConvGRU	+ConvLSTM	ResNet	+ConvGRU	+ConvLSTM
20	8.38	7.42(+0.04M)	7.28(+0.04M)	31.72	29.69(+0.04M)	29.81(+0.04M)
32	7.54	6.60(+0.06M)	6.88(+0.07M)	29.86	27.42(+0.07M)	28.11(+0.07M)
56	6.78	6.39(+0.11M)	6.45(+0.12M)	28.14	27.02(+0.11M)	27.26(+0.12M)

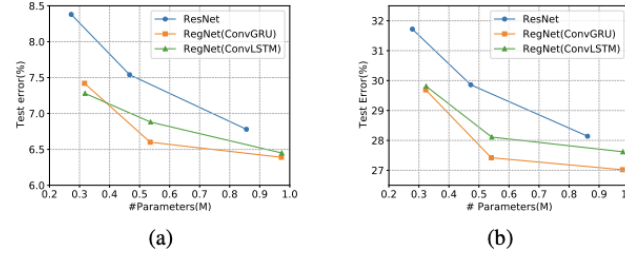


Fig. 4. Comparison of parameter efficiency on CIFAR-10 between RegNet and ResNet [5]. In both 4(a) and 4(b), the curves of our RegNet is always below ResNet [5] which show that with the same parameters, our models have stronger ability of expression.

규제 모듈을 추가함으로써 연산량이 늘어나긴 하지만 ResNet 32layer와 RegNet 20layer의 성능이 같다. 성능을 기준으로 연산량을 비교한다면 RegNet은 동일 성능 대비 ResNet 보다 파라미터를 획기적으로 감소시켰다.

## Positions of Feature Reuse

TABLE V  
TEST ERROR RATES ON CIFAR-10/100. WE USE CONVGRU AND CONVLSTM AS REGULATORS OF RESNET. WE LIST THE INCREASE OF PARAMETER THE ARCHITECTURES. IN EACH OF OUR REGNET<sub>(i)</sub> MODELS, THERE IS ONLY ONE CONVRNN APPLIED IN LAYER CONV<sub>i</sub>,  $i \in \{1, 2, 3\}$ .

model	C-10		C-100	
	err.	Params	err.	Params
ResNet [5]	8.38	0.273M	31.72	0.278M
RegNet <sub>(1)</sub> (GRU)	7.52	0.279M	30.40	0.285M
RegNet <sub>(2)</sub> (GRU)	7.48	0.285M	30.34	0.291M
RegNet <sub>(3)</sub> (GRU)	7.49	0.306M	30.30	0.312M
RegNet <sub>(1)</sub> (LSTM)	7.56	0.281M	30.23	0.286M
RegNet <sub>(2)</sub> (LSTM)	7.49	0.290M	30.28	0.296M
RegNet <sub>(3)</sub> (LSTM)	7.52	0.325M	29.92	0.331M

이 표는 Conv RNN을 ResNet에 몇 번째 레이어에 부착했는지에 따른 표이다. Conv RNN은 딱 한 개만을 사용했으며 결과에 따르면 low layer에 부착하는 것이 적은 연산량으로 보다 큰 효율을 가져 올 수 있었다.

## RESULT

우리는 Conv RCNN을 통해 ResNet의 성능을 증가시킬 수 있는 규제 모듈을 제안했다. 성능이 증가되는 것을 3가지 데이터 셋으로 입증했다.