ELLab

SUNG KYUN KWAN UNIVERSITY

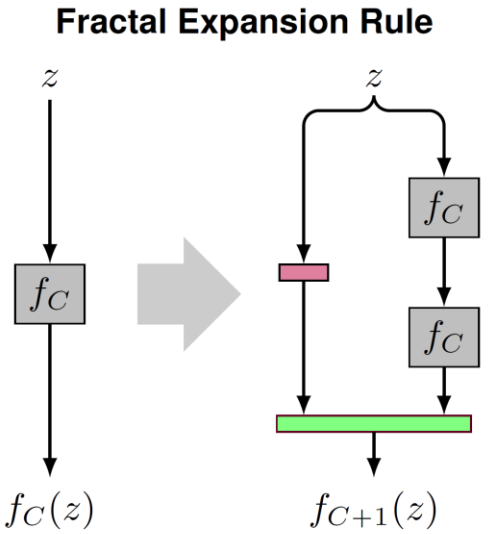
FractalNet

2025.11.28

Introduction

FractalNet

- Self similarity에 기반한 신경망 → 확장하다보면 truncated fractal 형태가 됨
- 길이가 서로 다른 sub path의 상호작용 (not residual connection)
- 네트워크 얕은 구조 → 깊은 구조 효과적 전환



ResNet

- Residual을 학습함
- 대부분 layer들이 identity처럼 동작 → loss까지 거리를 효과적으로 줄여줌

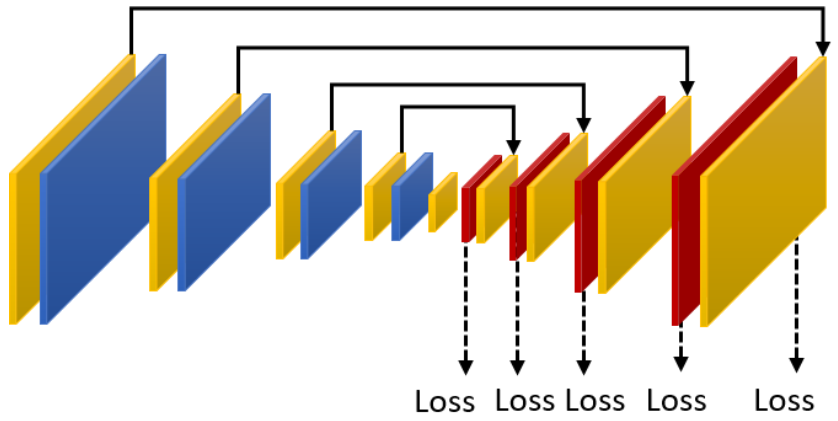
FractalNet

- Residual 필수는 아님
- Deep supervision과 student-teacher 학습형태 유도함
- Simplicity of training (학습의 단순함 ↔ 설계의 단순함)
- 여러 깊이의 sub-network들을 포함 → 전반적인 깊이에 대부분 Robust함

Introduction

Deep Supervision

- 중간 layer들에 loss를 추가해서 네트워크의 다양한 깊이에서 직접적으로 학습 신호를 전달함
- FractalNet에서는 얕은 경로부터 깊은 경로까지 다양한 subnetwork들이 존재함
- 여러 깊이의 subnetwork 들이 같은 입력과 같은 최종 loss를 공유하고 있음
- 얕은 column의 초반 conv 레이어들은 마치 중간 레이어까지 classifier를 붙여 놓은 deep supervision처럼 강한 gradient를 직접 받게 됨 -> 학습 쉬워짐
- 중간층이 직접 감독 받는 것과 비슷한 효과가 자연스럽게 생김

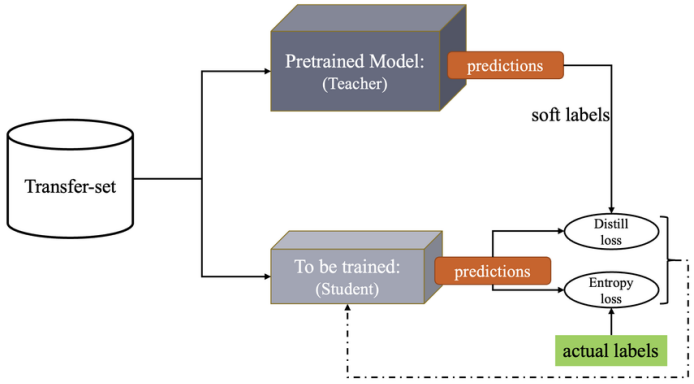


Introduction

Student – Teacher Learning

- Teacher(큰 네트워크)를 먼저 학습시키고 Teacher의 soft output을 Student(작은 네트워크)가 따라하게해서 Student 성능을 끌어올리는 방식
- Subnetworks: deep path → student, shallow path → teacher
  - Shallow path는 capacity가 낮아서 처음엔 빠르게 수렴가능
  - Deep path는 gradient 흐름이 길어 초기 학습이 매우 어려움
- Join(element wise mean)
  - Join에서 element wise mean 하기 때문에 깊은 네트워크가 이상한 값을 내면 전체 출력이 나빠지고 loss가 커짐
  - 깊은 네트워크도 점점 얕은 네트워크가 주는 출력 분포를 따라가도록 학습됨
- Drop-path
  - 어떤 미니배치에서는 얕은 path가 drop되고 깊은 path만 남음
  - 깊은 것 혼자서도 제대로 예측해야 하므로 student의 독립적인 능력을 키움

"Both shallow & deep subnetwork must individually produce correct output"



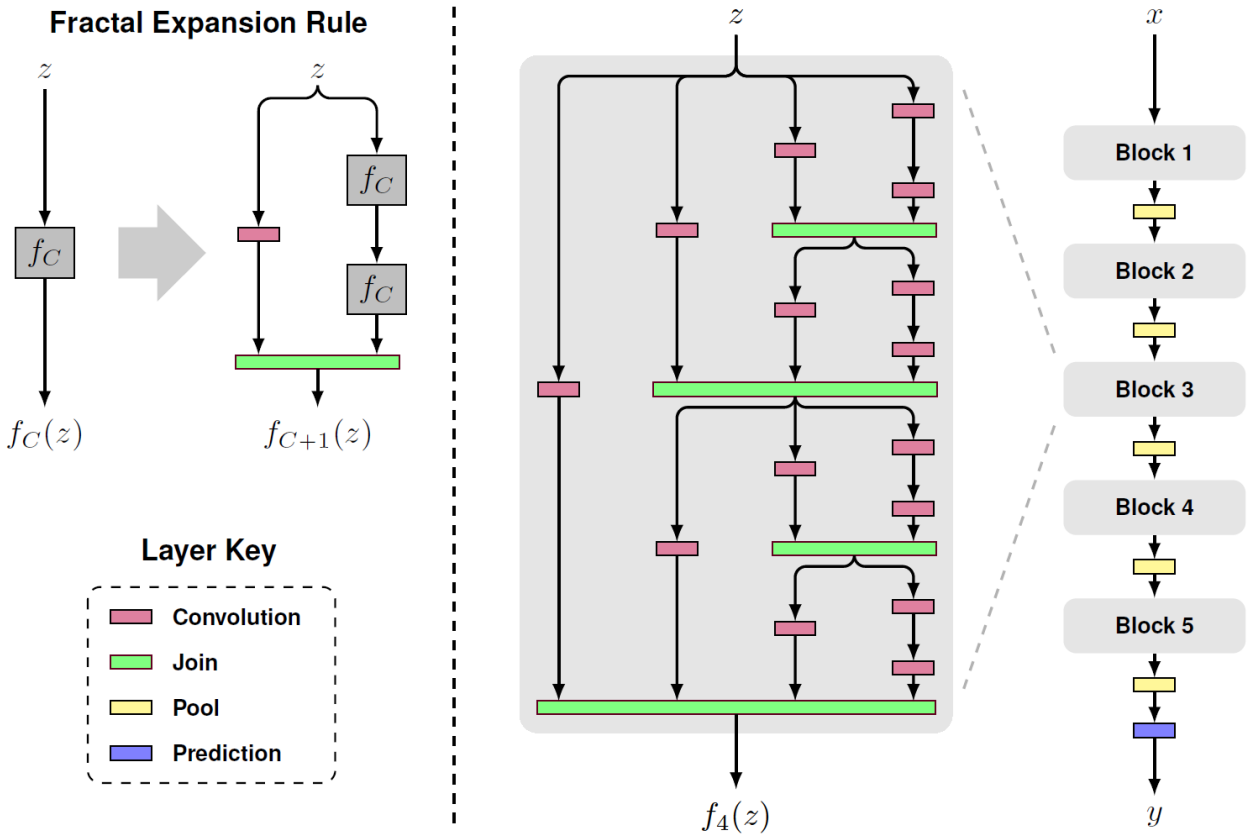
Fractal Network

Architecture

$$f_1(z) = \text{conv}(z)$$

$$f_{C+1}(z) = [f_C \circ f_C(z)] \oplus [conv(z)]$$

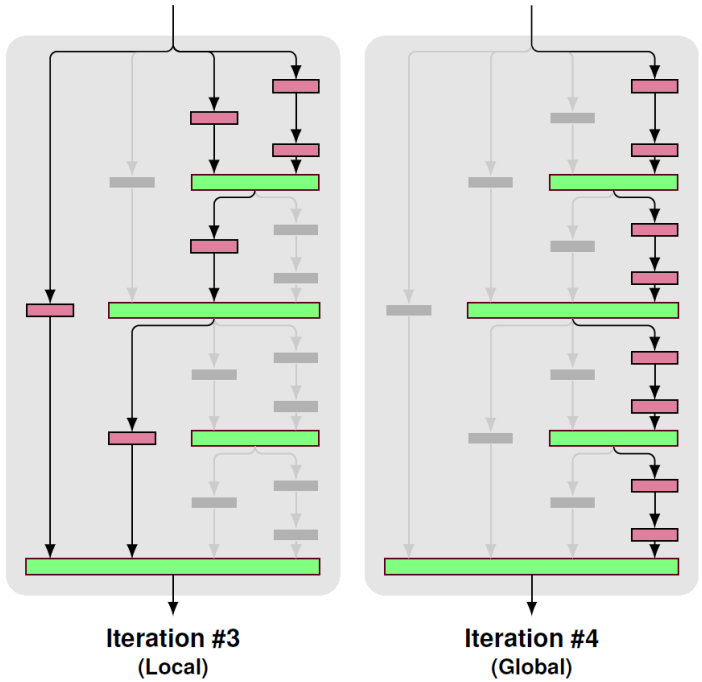
- C: # of columns, or width
- 2^{C-1}: # of conv layers on the longest path
- B: # of building blocks
- $B \cdot 2^{C-1}$: total depth
- $\oplus$: join(element wise mean)



Fractal Network

Drop-path

- Local sampling
  - 각 join layer마다 독립적으로 path를 drop
  - 확률적으로 각 join에서 몇 개 path가 사라지지만 최소 하나는 반드시 남김
- Global sampling
  - 전체 Fractal block에서 단일 column을 전체 mini-batch 동안 선택
  - 독립적으로 강한 단일 column이 되도록 학습시킴
- Effects
  - Regularization(overfitting, co-adaptation 방지), Ensemble-like, Training stability
- Training
  - Dropout + Drop-path
  - Drop-path는 local 50% + global 50% mixture model



Experiment

Results

- Table 2
  - FractalNet-34는 ResNet-34와 거의 동일한 정확도를 냄
  - 둘 다 VGG보다 훨씬 좋음
  - FractalNet도 ResNet처럼 deep architecture 대열에 들어갈 정도의 성능을 가짐

Method	Top-1 (%)	Top-5 (%)
VGG-16	28.07	9.33
ResNet-34 C	24.19	7.40
FractalNet-34	24.12	7.39

Table 2: **ImageNet** (validation set, 10-crop).

- Table 3
  - 깊이가 깊어질수록 성능이 좋아짐 (80까지)
  - 그러다 160에서는 살짝 나빠지지만 학습이 완전 붕괴되진 않음

Cols.	Depth	Params.	Error (%)
1	5	0.3M	37.32
2	10	0.8M	30.71
3	20	2.1M	27.69
4	40	4.8M	27.38
5	80	10.2M	26.46
6	160	21.1M	27.38

Table 3: **Ultra-deep fractal networks**

Experiment

Results

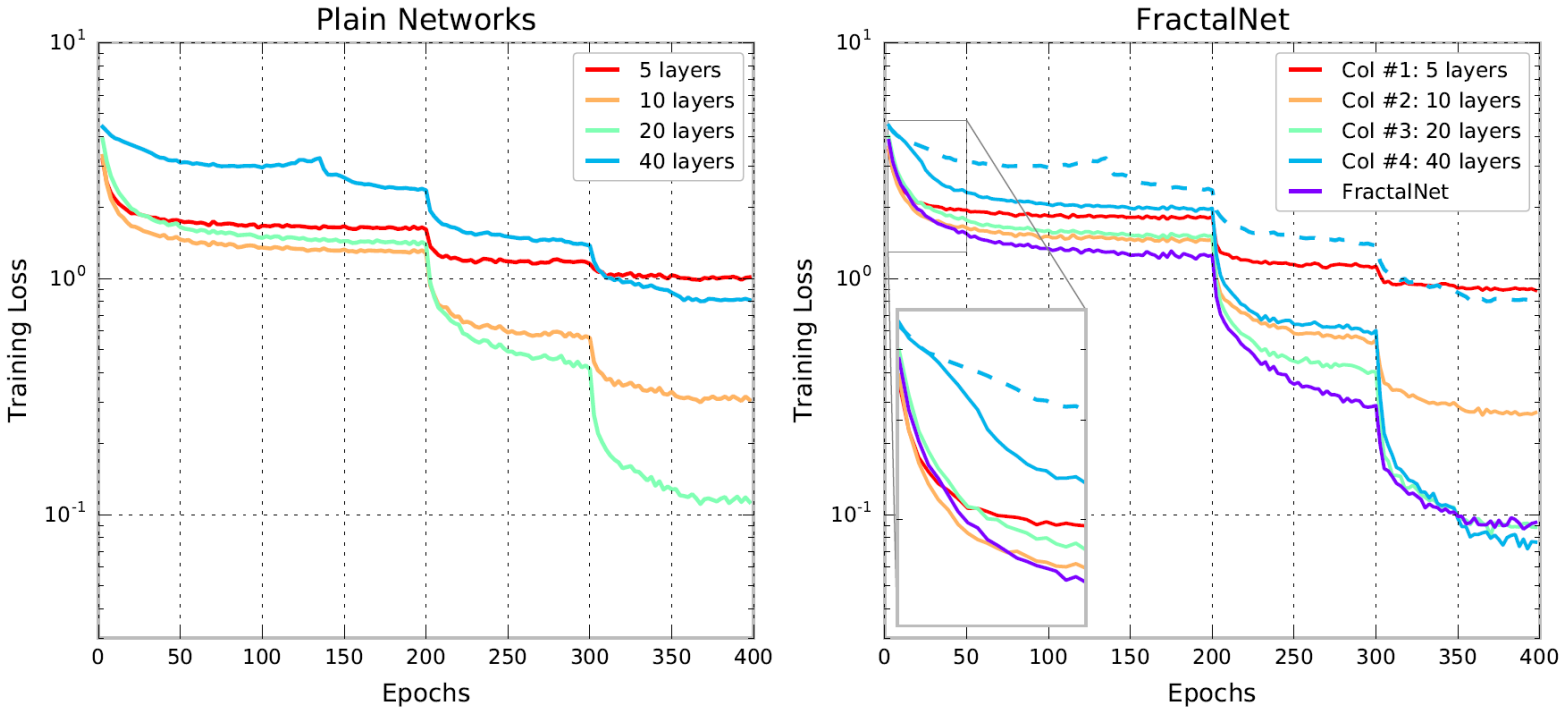
- Table 4
  - Plain(40)의 train loss = 0.580 → 급격히 나빠짐
  - Plain network는 깊이가 40이 되면 gradient vanishing/exploding 때문에 학습이 붕괴됨
  - Fractal column 4(40)은 훨씬 정확함
  - Fractal 구조가 깊은 네트워크를 안정적으로 학습시키는 효과가 있음

Model	Depth	Train Loss	Error (%)
Plain	5	0.786	36.62
Plain	10	0.159	32.47
Plain	20	0.037	31.31
Plain	**40**	**0.580**	**38.84**
Fractal Col #1	5	0.677	37.23
Fractal Col #2	10	0.141	32.85
Fractal Col #3	20	0.029	31.31
Fractal Col #4	**40**	**0.016**	**31.75**
Fractal Full	**40**	**0.015**	**27.40**

- student–teacher effect
  - Shallow net = teacher
  - Column #4 = student
  - shallow columns가 initial teacher 역할 해주면서 깊은 네트워크도 비교적 안정적으로 수렴함

Experiment

Result



- Implicit deep supervision (Elbow-shaped curve)
  - shallow columns이 먼저 안정화
  - drop-path로 인해 deep column도 이 경로를 따라 signal을 받게 됨
  - teacher-like pressure가 생기면서 deep column이 학습됨

My Code

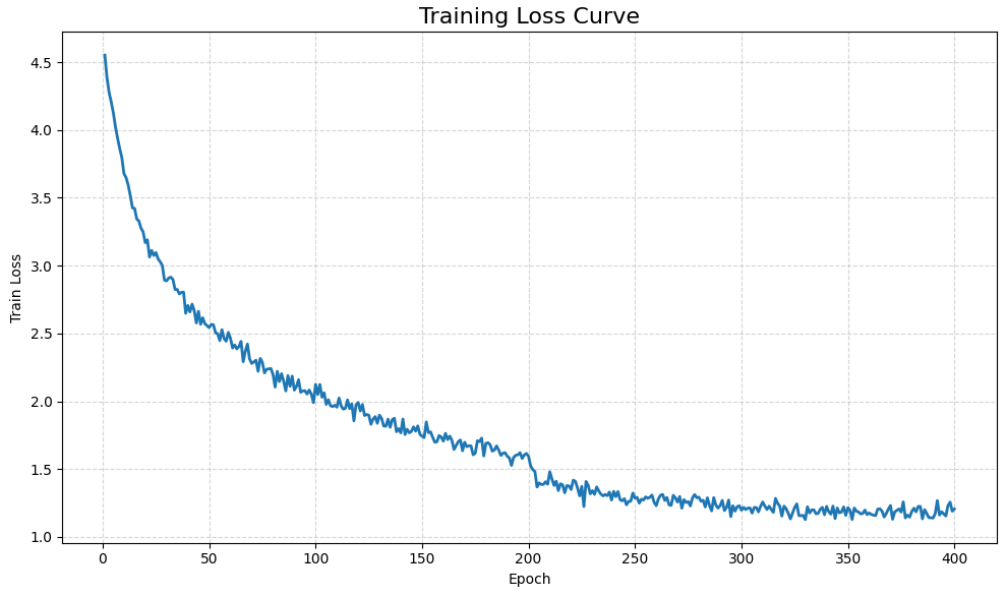
학습 세팅 by implementation details

항목	논문/구현 설정값	항목	논문/구현 설정값
데이터셋	CIFAR-100 (45k train / 5k val)	**Optimizer**	SGD (momentum = 0.9)
이미지 전처리	32×32, 4픽셀 zero-padding → 32×32 random crop, horizontal flip	**Batch size**	100
블록 수 (B)	5	**Weight decay**	1e-4
블록별 채널 수	(64, 128, 256, 512, 512)	**초기 Learning rate**	0.02
블록 깊이 (columns)	C = {1,2,3,4} → 깊이 {5, 10, 20, 40}	**Learning rate schedule**	epoch 200, 300, 350에서 ×0.1 감소
Pooling	각 블록 사이마다 2×2 max-pooling + subsampling	**Epochs**	400
Dropout (block dropout)	블록별 drop rate = {0%, 10%, 20%, 30%, 40%}	**Weight Initialization**	Xavier initialization
Drop-path (local)	네트워크 전체에 drop-rate = 15%	**Drop-path sampling**	50% local, 50% global (mini-batch 단위)

My Code

Experiment

- CIFAR 100인걸 감안해도 예상보다 낮은 정확도를 보임
- Local drop-path할때 최소 한 경로는 살려야 한다는 조건을 둘다 drop 될 경우 shallow만 살리는 조건문으로 구현했더니 loss가 더 풍부한 표현을 하지 못하고 멈춰버린 것 같음
- 애초에 Global drop하는 코드도 잘못 구현함
- 다른 사람 코드 보면 mask로 구현하던데 직관적으로 어떻게 쉽게 구현할 수 있을지 모르겠음

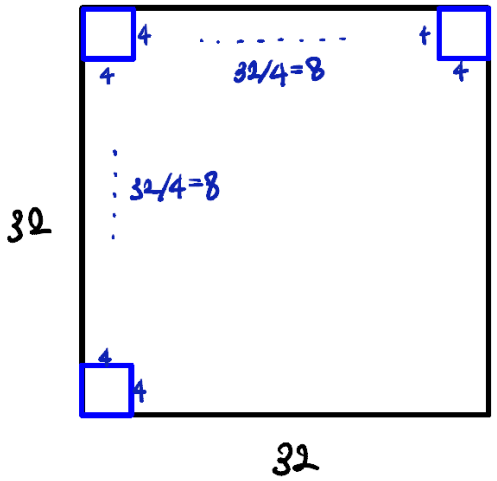


https://colab.research.google.com/drive/1CxyjPHwJVFNgUxM98v2R80eaHqMabEDR?usp=sharing

Columns	Depth	score
4	40	49.62%

My Code - Revised

FractalNet

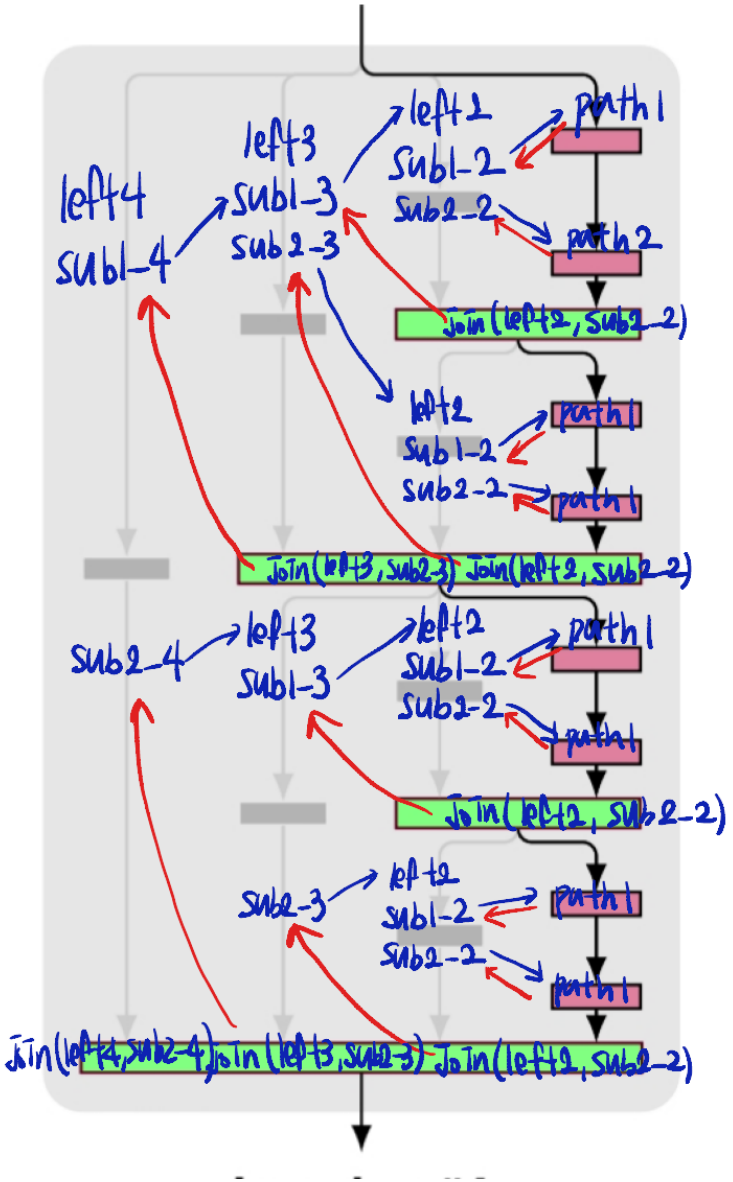


Input [B,3,32,32] → embedding [B,48,64]

$N_p = 8 \times 8 = 64$

embedding size per patch = $4 \times 4 \times \underbrace{3}_{\text{channel}} = 48$

https://colab.research.google.com/drive/1CxyjPHwJVFNgUxM98v2R80eaHqMabEDR?usp=sharing



My Code - Revised

FractalNet

model	score	Dataset
Plain(40)	83.42%	CIFAR10
FractalNet(40)	89.22%	CIFAR10
FractalNet(40)	72.86%	CIFAR100

ELLab