Denoising Monte Carlo Images with Machine Learning

Team 5: Cheolmin Lee, Minki Jo, Nick Heppert

INDEX

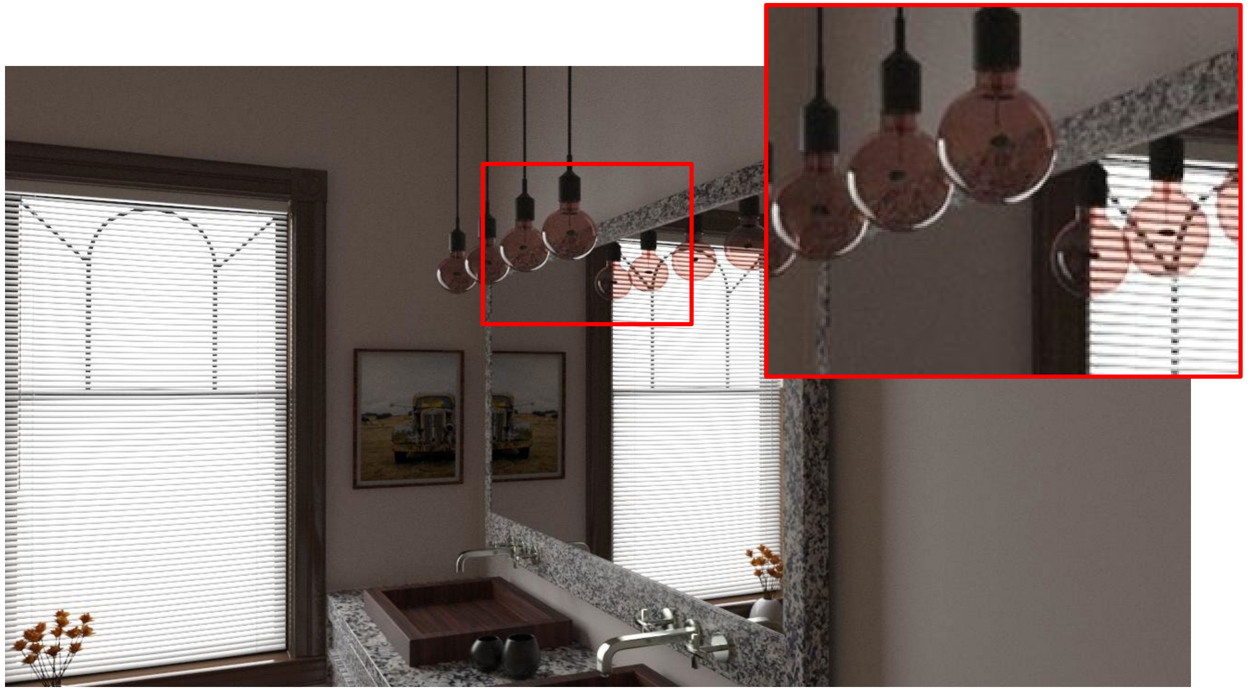
Introduction

1. Problem
2. Machine Learning & Deep Learning

Introduction – Problem

- Creating images with high samples per pixels (spp) takes a lot of time
- Cut down time by creating low samples images → Noisy
- De-Noising techniques

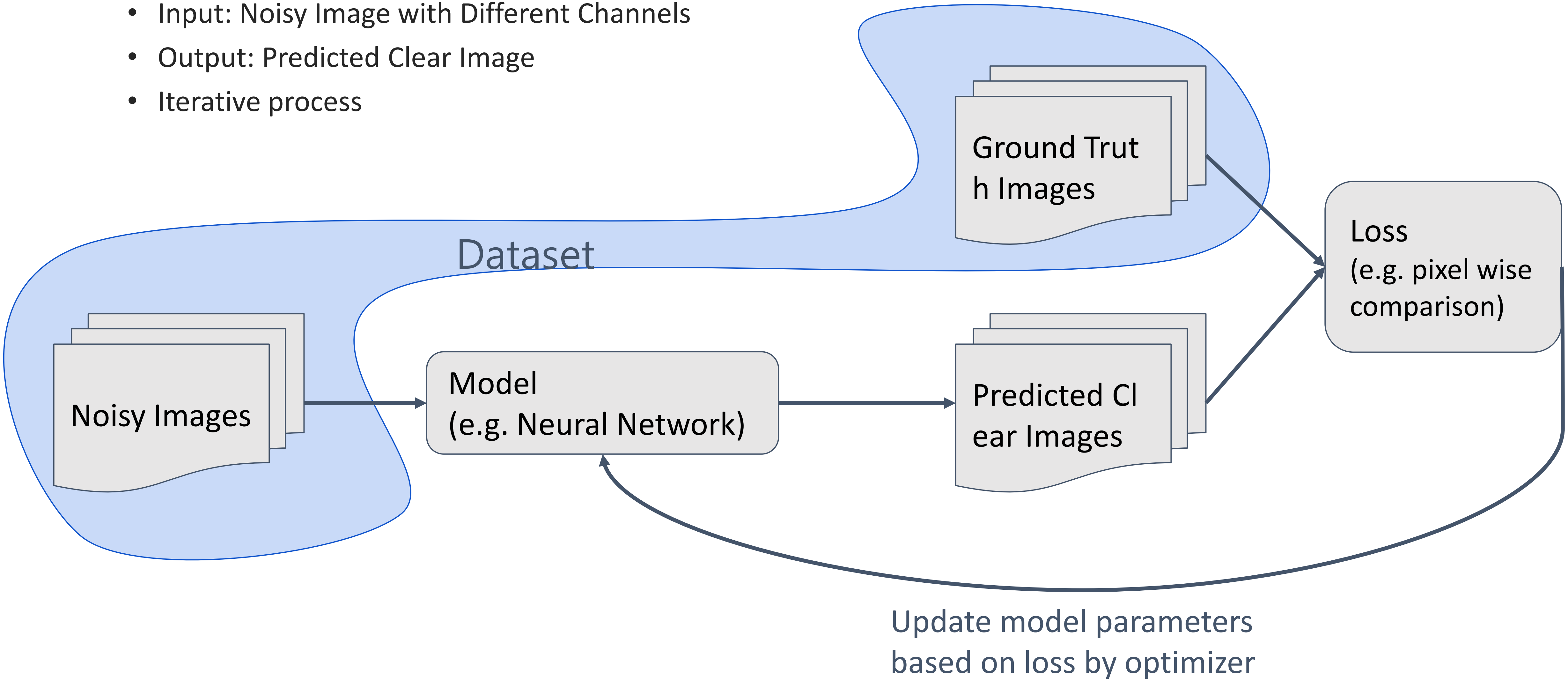
128spp



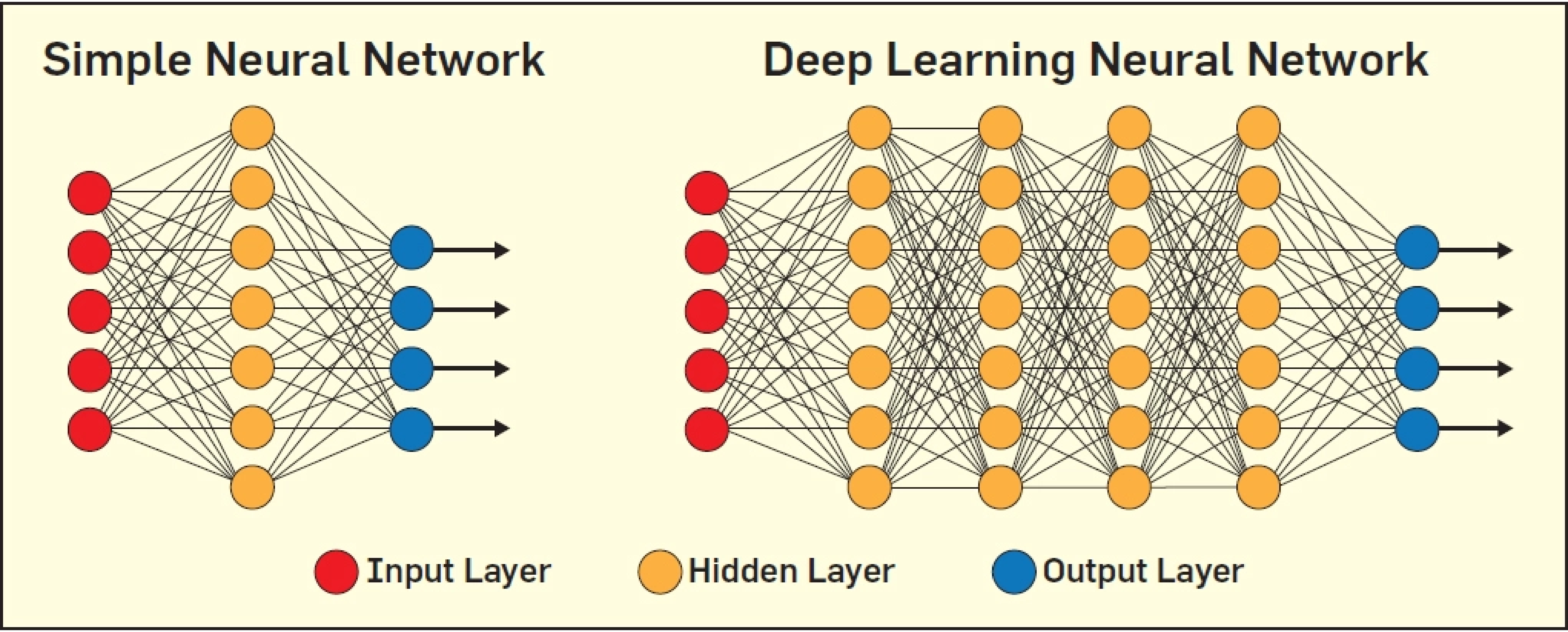
8192spp

Background: Machine Learning

- Supervised Setup
  - Input: Noisy Image with Different Channels
  - Output: Predicted Clear Image
  - Iterative process

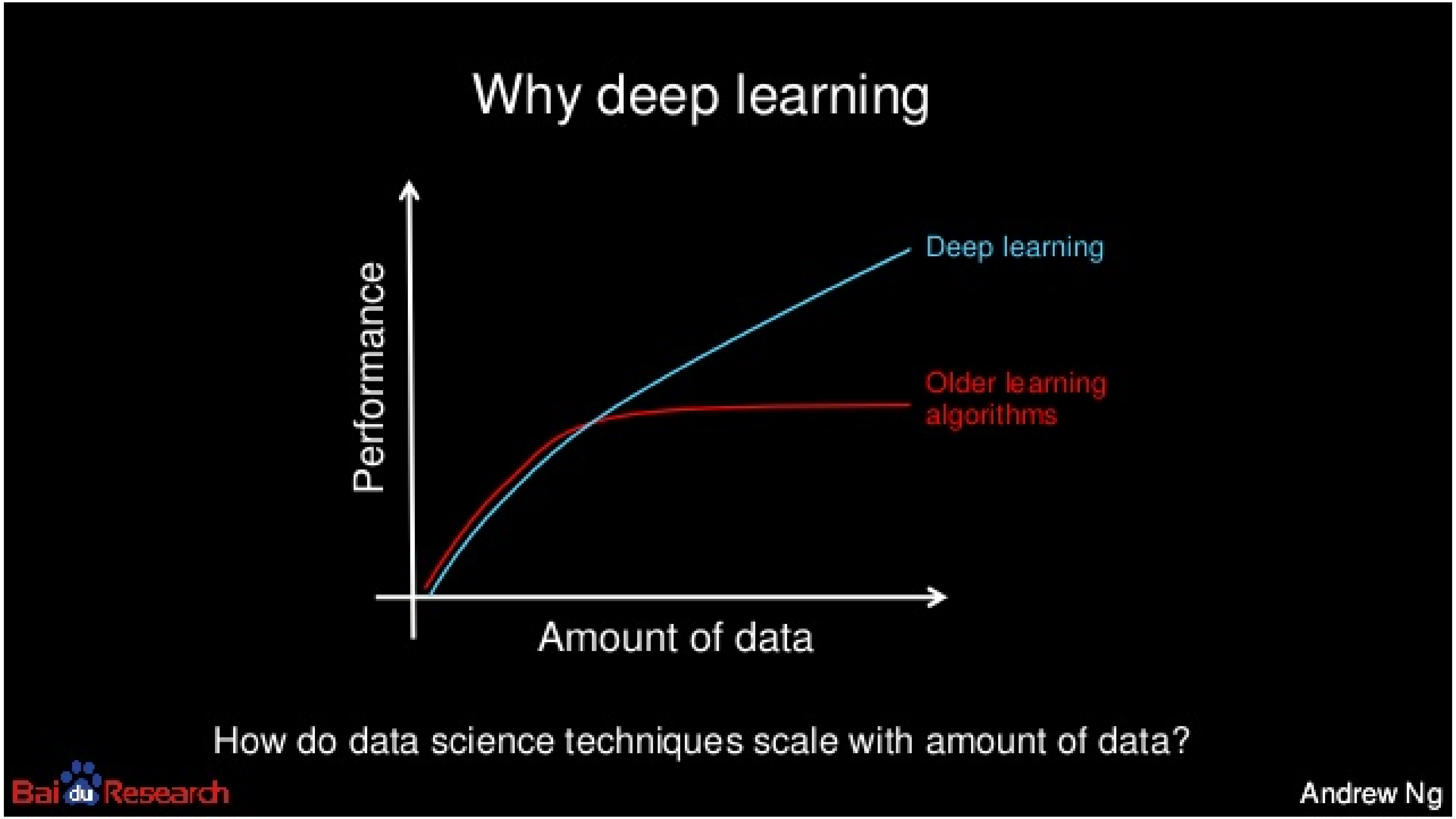


Background: Deep Learning



https://cacm.acm.org/magazines/2018/6/228030-deep-learning-hunts-for-signals-among-the-noise/

Background: Deep Learning



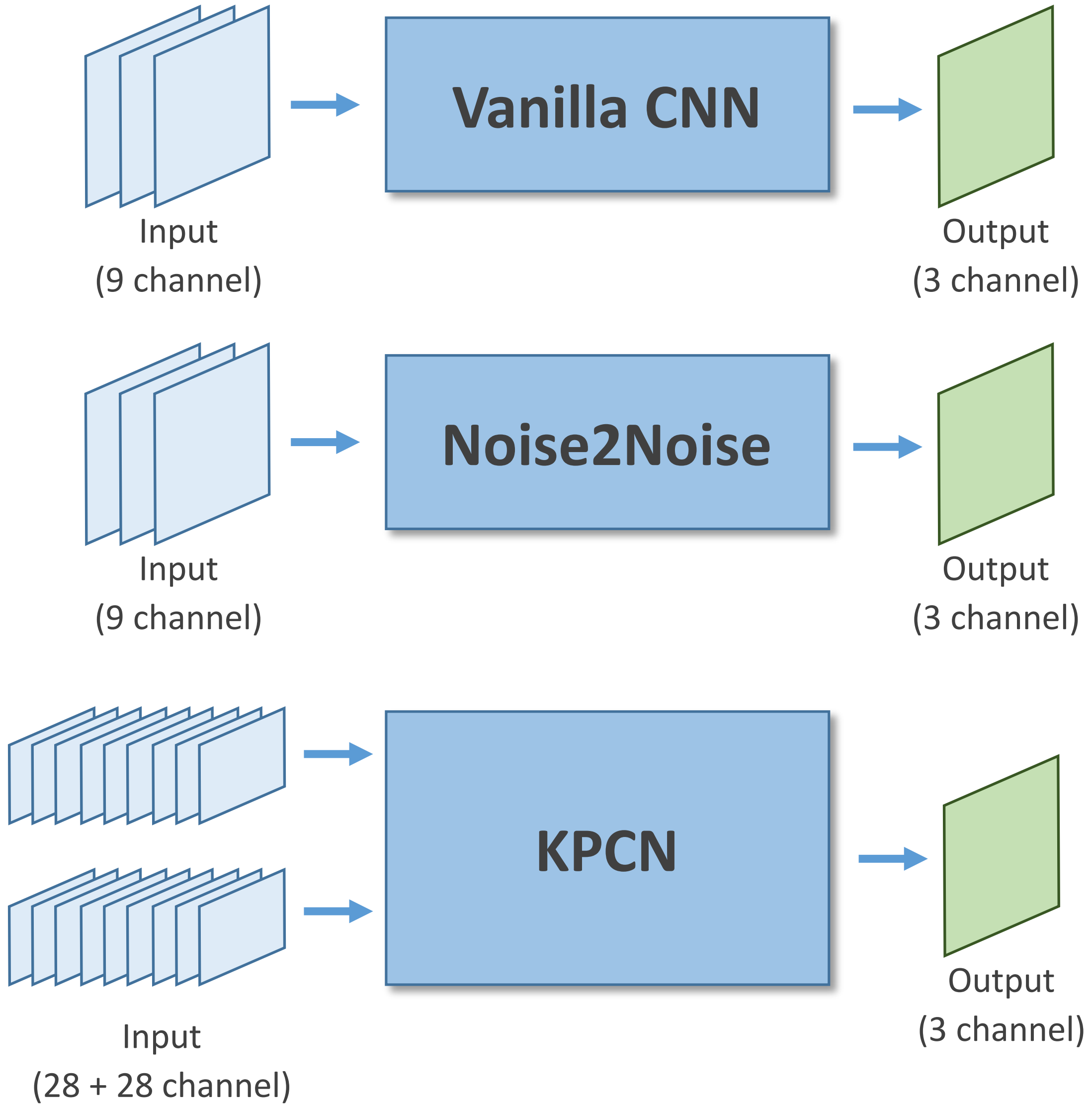
https://www.slideshare.net/ExtractConf/andrew-ng-chief-scientist-at-baidu/30

Problem
&
Approach

Problem

Problem of KPCN

- KPCN takes 28 channel of input for each diffuse and specular network.
- Considering other 3D rendering denoising models, KPCN needs high-dimensional input data.

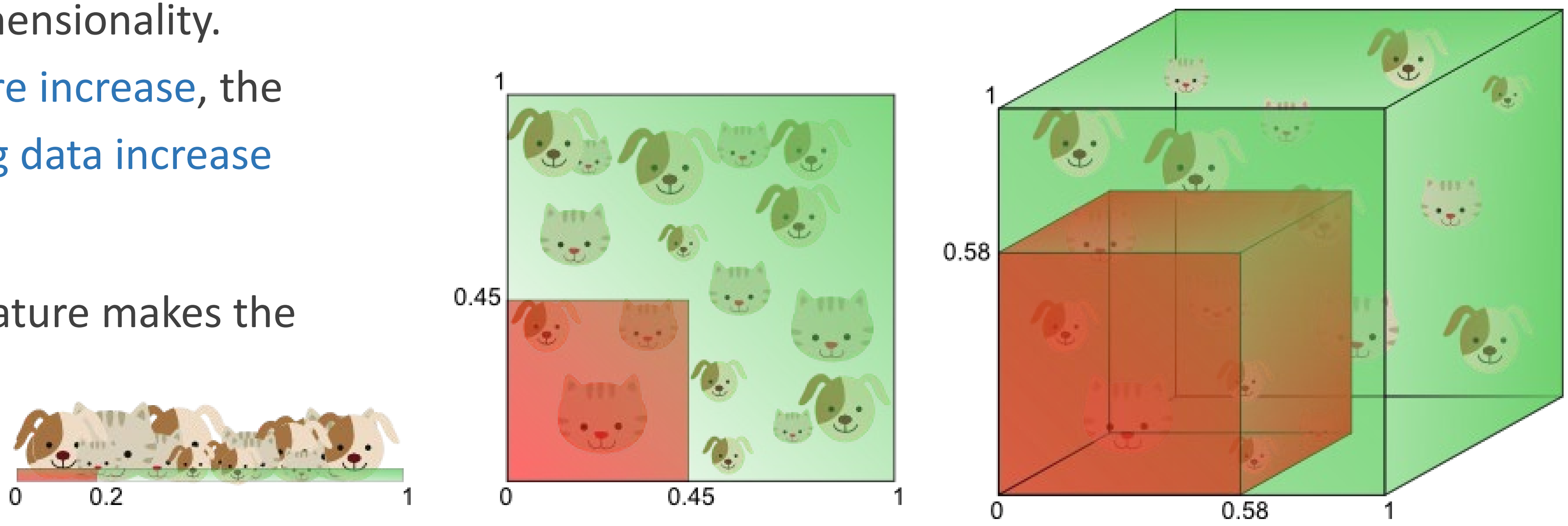


KPCN은 diffuse, sepcular 네트워크에 각각 28채널의 인풋을 받는다.
다른 디노이징 모델들은 일반적으로 9채널 인풋을 사용하는 것을 생각하면, KPCN이 상당히 많은 데이터를 사용하는 것이다.

Problem

Why is this a problem?

- Because of the curse of dimensionality. When the number of feature increase, the amount of required training data increase exponentially.
- In other word, too many feature makes the training hard.



Let's say you want to fill 20% of the space.
If you use only one feature, 20% of the data in feature 1 is required.
When you use two feature, 45% of the data for each features is required.
When you use three feature, 58% of the data for each features is required.

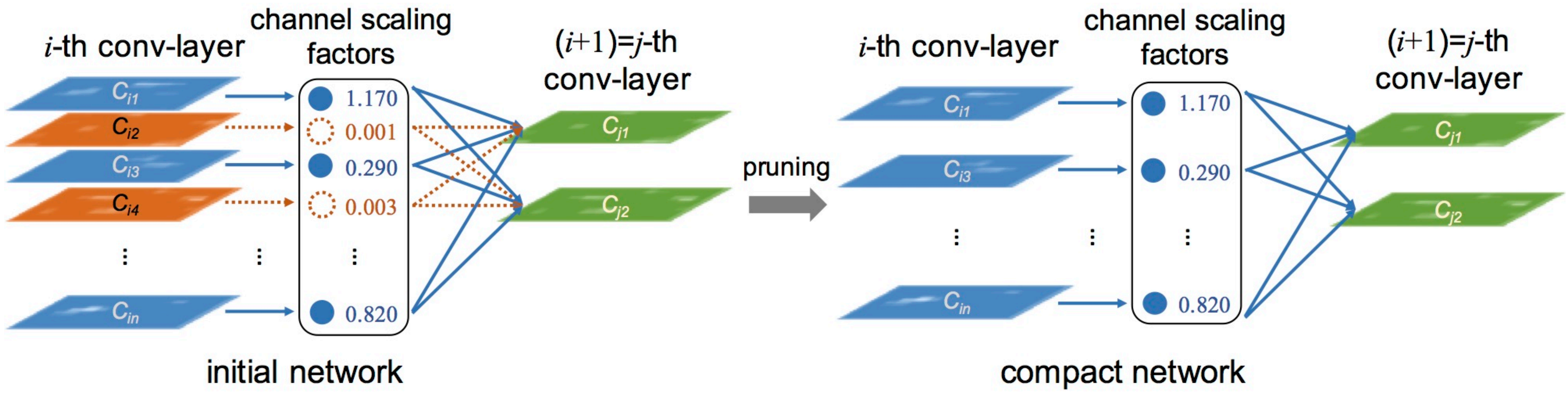
Feature가 많아질 때마다 필요한 training 데이터 수가 급격히 증가한다.
즉 많은 Feature를 사용하면 학습이 매우 어려워진다.

Motivation – Pruning

Problem of Pruning

- Most of the variables in Neural network are highly correlated with other variables. When you prune the 70% of the variables, the performance of the model does not change or even gets better.

네트워크 내부에는 매우 유사한 variable이 반복되어 나타나는 경우가 많다.
이런 불필요한 variable들을 삭제해서 필요한 변수만 남기면 계산이 빨라지고 성능이 개선된다.



[Liu et. al. 17] Learning Efficient Convolutional Networks through Network Slimming, ICCV2017

Motivation – Pruning

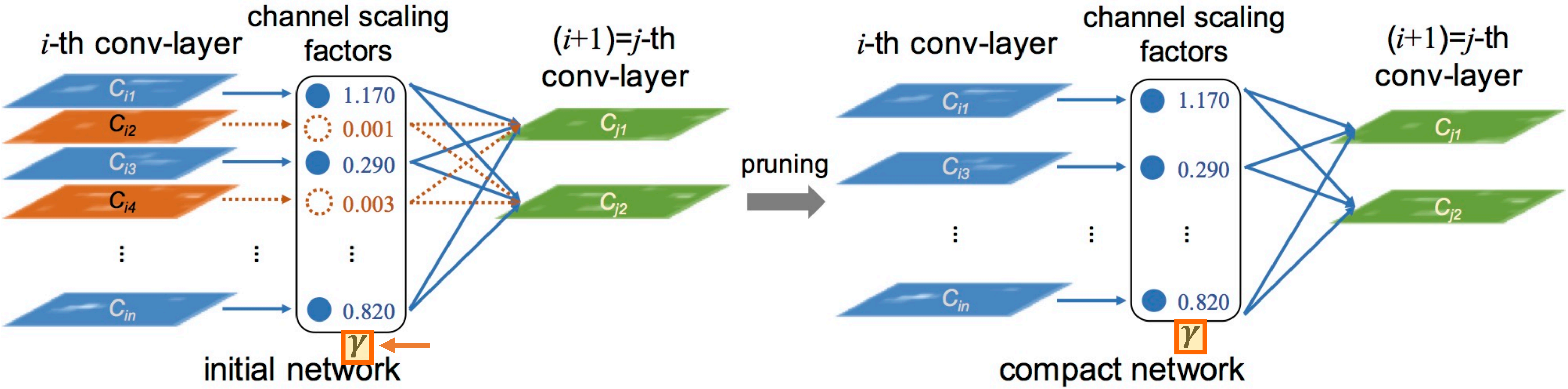
$$\text{Loss} = \text{Accuracy} + \boxed{|\,\gamma\,|}$$

When the gamma represent the importance

Method of pruning

- When you regulate the sum of the importance value of the variables by loss, only necessary (important) variable remain and redundant variable goes to 0.

네트워크의 중요도를 나타내는 변수의 총합을 로스로 제공하면
중요한 variable들은 남고 불필요한 변수의 값은 0에 가까워진다



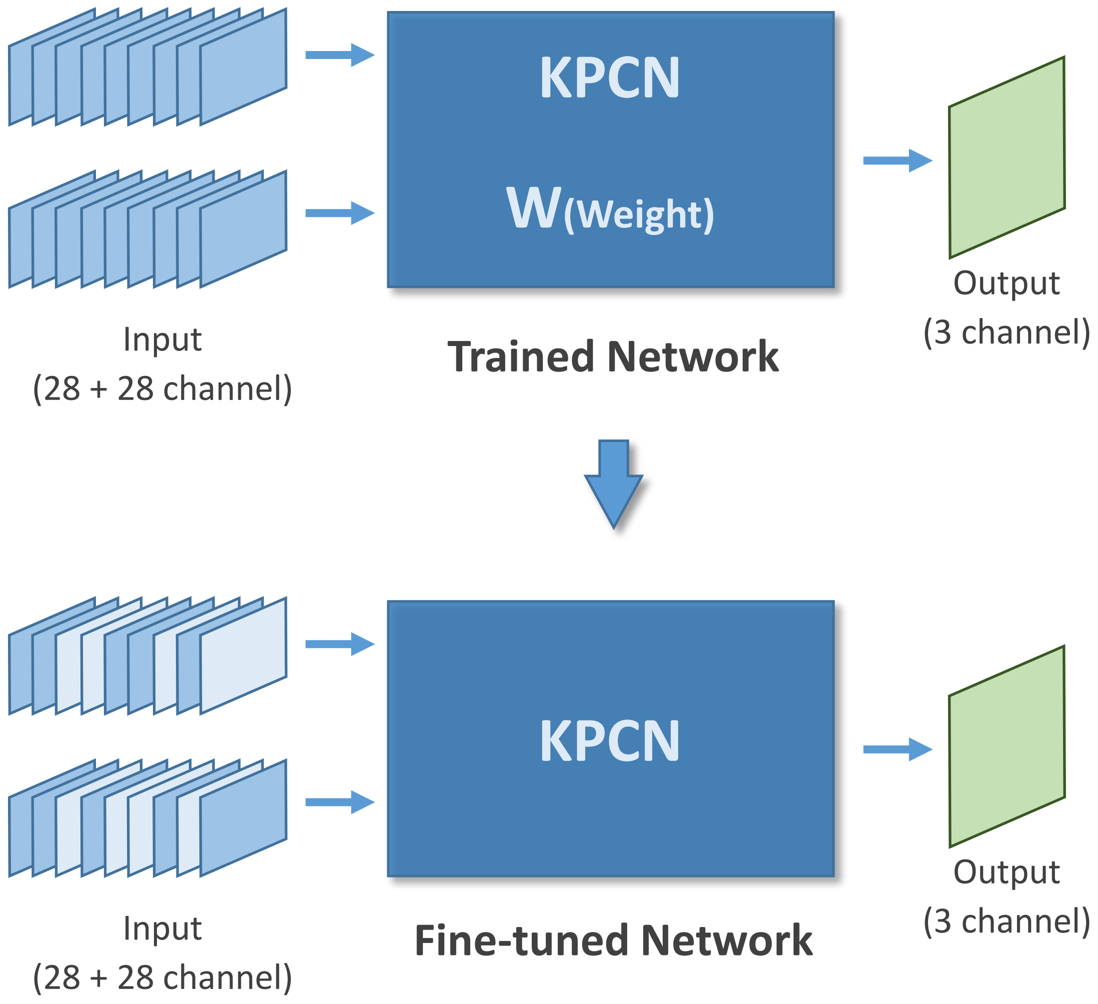
[Liu et. al. 17] Learning Efficient Convolutional Networks through Network Slimming, ICCV2017

Method

Input channel Pruning

- The size of the weight value itself is also one of the representative value of the importance.
- If we regulate the total weight value of the first layer, or apply other pruning technique, we can find the less important input channel.

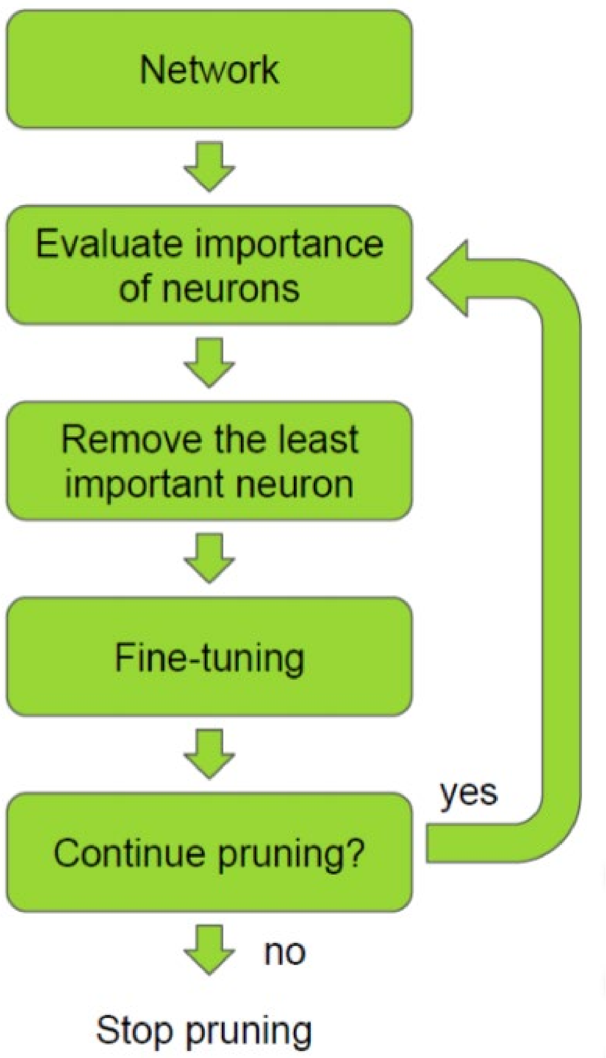
Loss = Accuracy + $\boxed{| \, W \, |}$

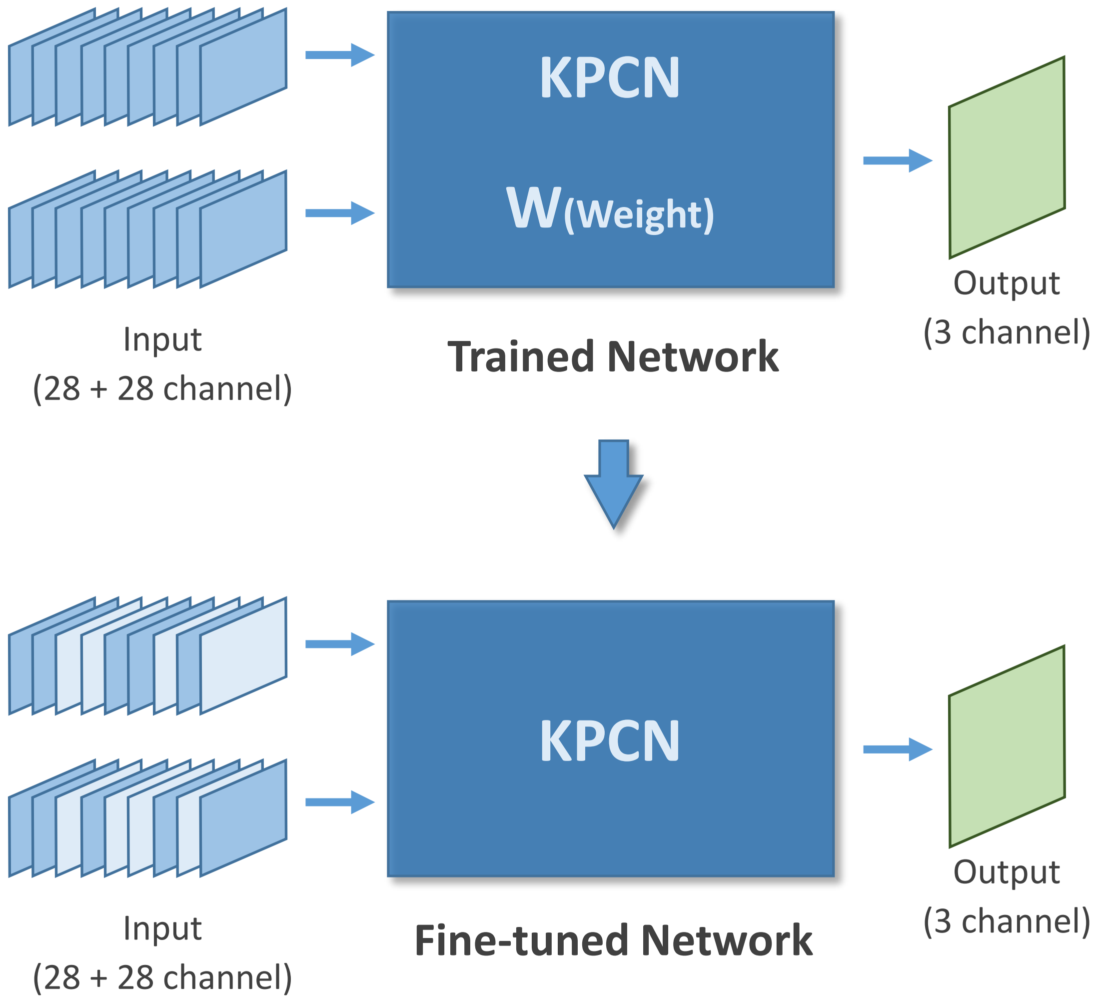


네트워크의 웨이트값 또한 중요도를 나타낸다. 따라서 첫 레이어의 웨이트 총합을 제한하여 학습하면 중요도가 낮은 채널을 찾아낼 수 있다.

Method

Learning Algorithm

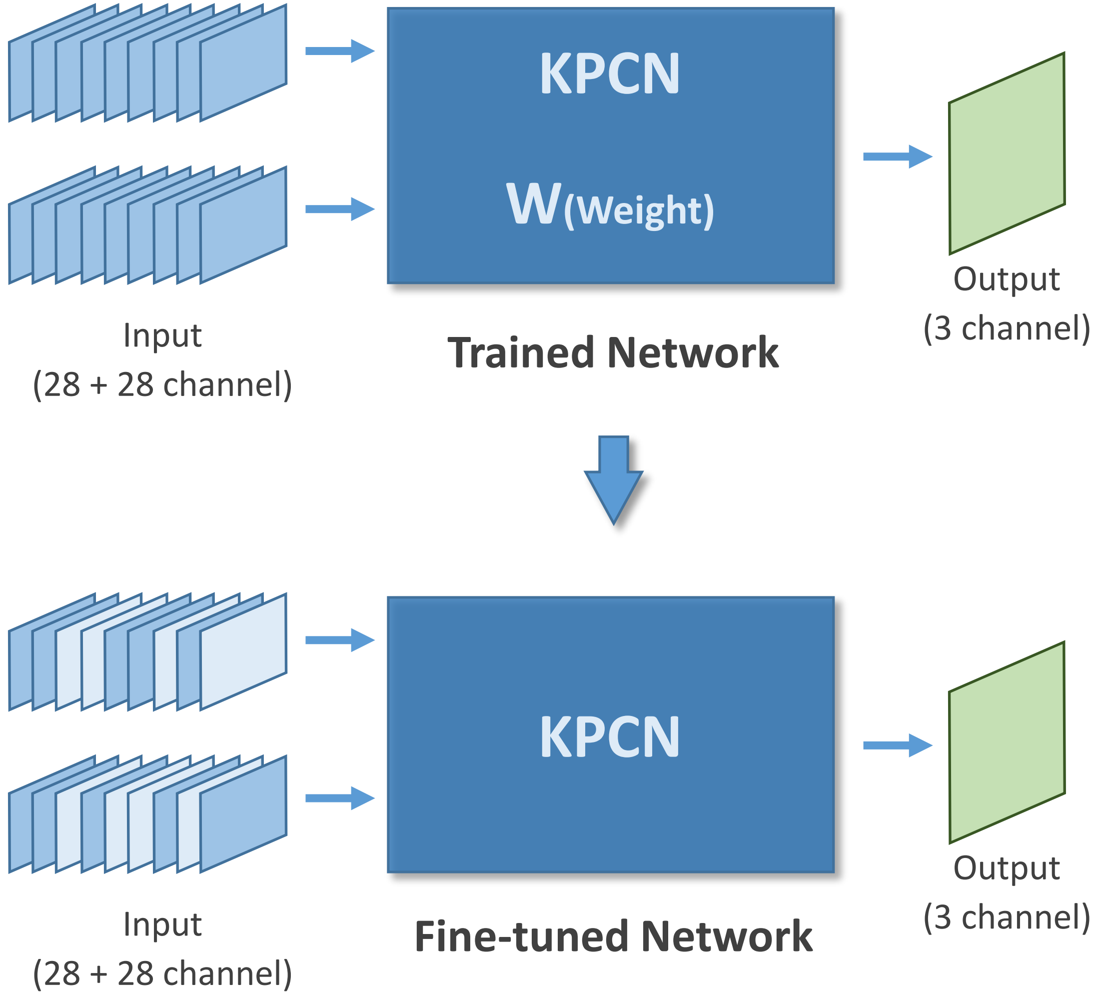




Method

Expected effect

- Faster training
- Improvement of the performance
  by reducing the dimension of the problem



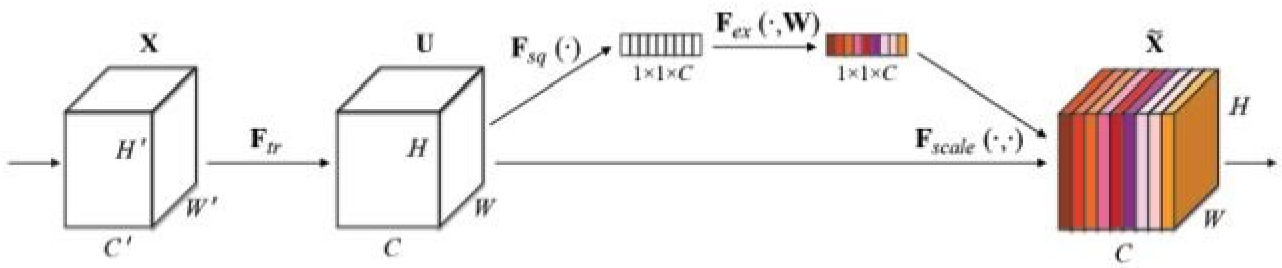
Loss = Accuracy + | W |

기대되는 효과는 빠른 학습 및 문제의 차원 감소로 인한 성능 개선이다.

Minor idea

Recent novel network architectures

- Channel attention
  This method can break the high correlation between channels and improve the performance of the model.



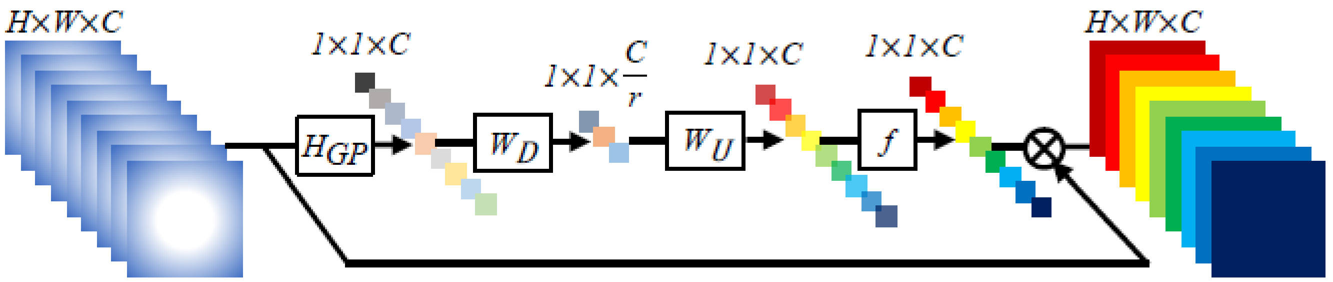


Fig. 3. Channel attention (CA). ⊗ denotes element-wise product

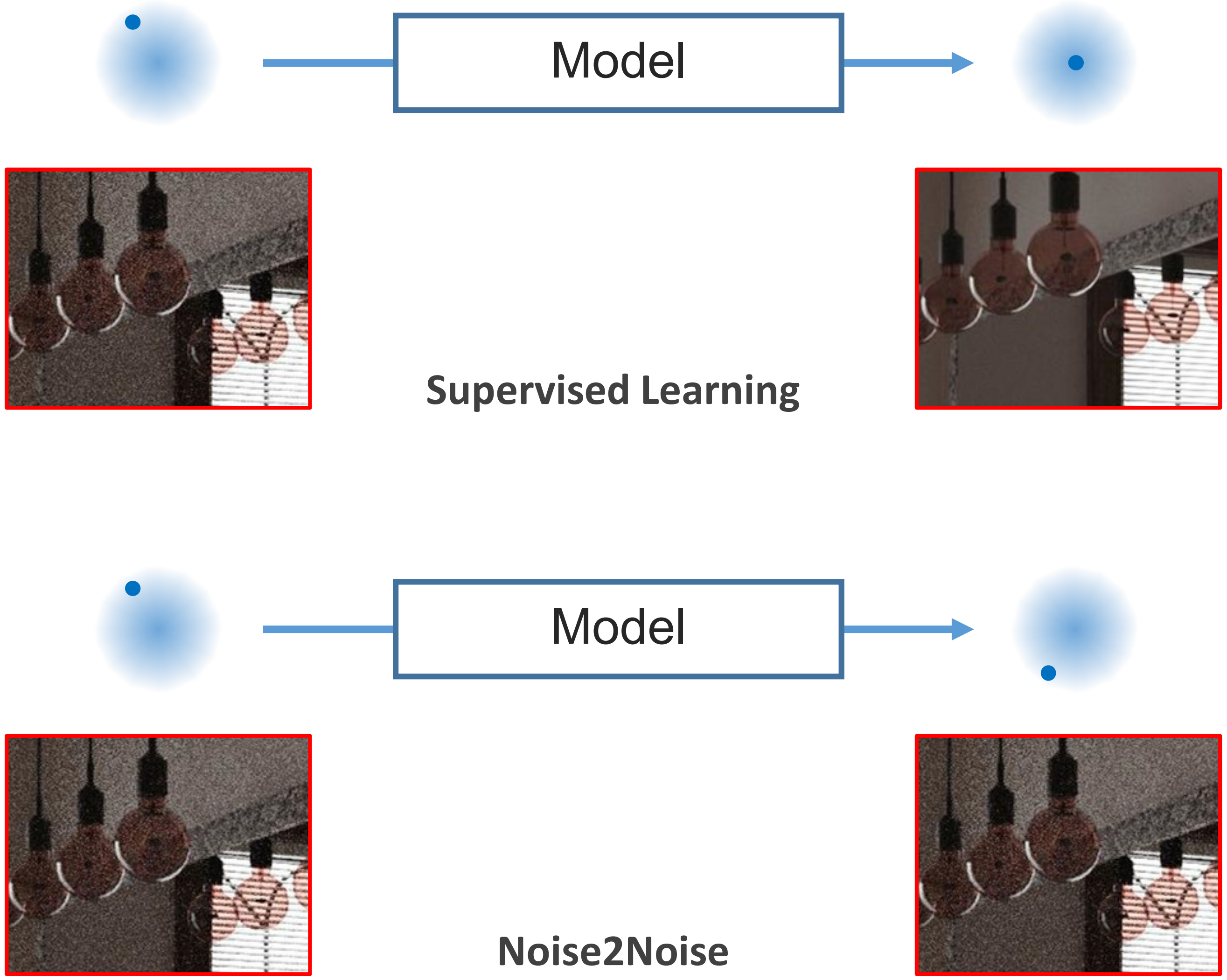
[Zhang et. al. 18] Image Super-Resolution Using Very Deep Residual Channel Attention Networks, ECCV2018

Additional Approach

Additional Approach

- Noise2Noise
  N2N is the current state of the art model for the single RGB denoising problem.

- We will try to merge N2N and KPCN model if we have enough time.



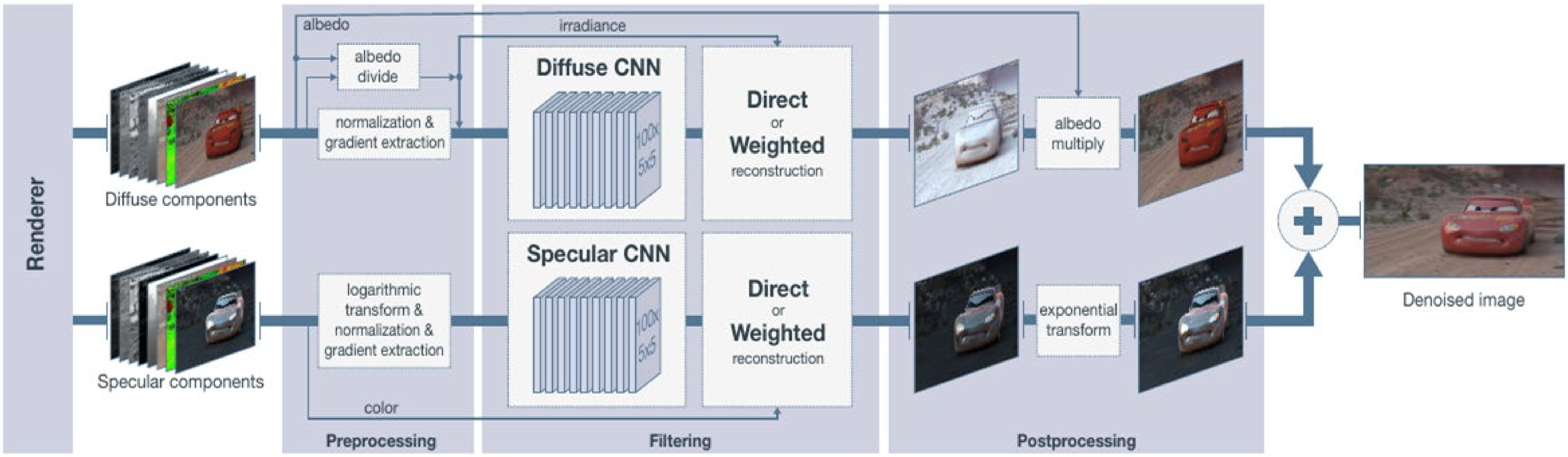
[Lehtinen et. al. 18] Noise2Noise: Learning Image Restoration without Clean Data, ICML2018

Progress & Future Plan

1. Current status
2. Schedule

Baseline: KPCN

- Kernel-predicting convolutional networks (KPCN) for denoising Monte Carlo renderings
  - Uses 56 (28 + 28) channels, Patches of size 64x64
- Network requires a lot of data to train

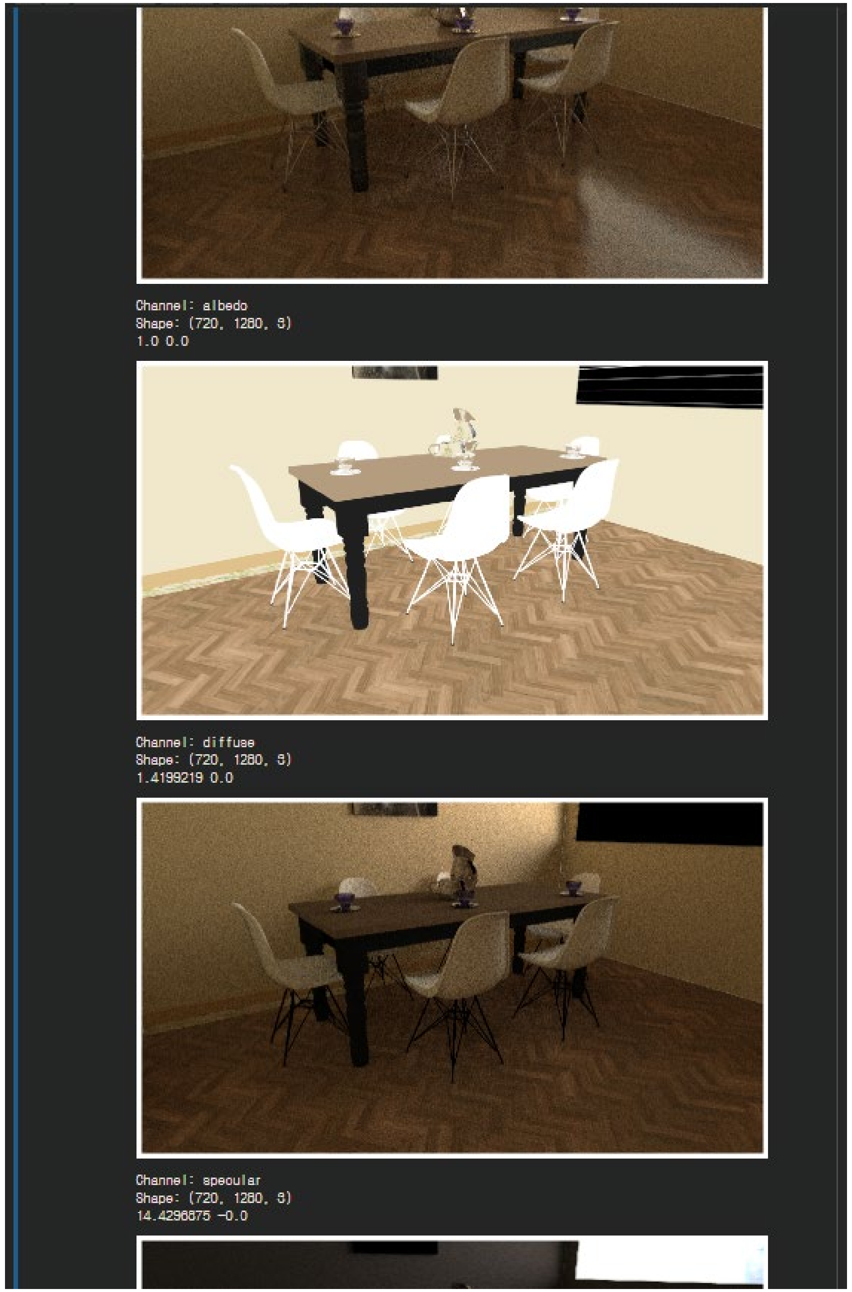


Bako Et al., "Kernel-Predicting Convolutional Networks for Denoising Monte Carlo Renderings." ACM Transactions on Graphics 36, no. 4 (July 20, 2017): 1–14. https://doi.org/10.1145/3072959.3073708.

Current Status

Plan

- Training dataset acquire **Done**
- Reproduction **Done**
- Porting
  - Model Manager **Done**
  - Training Monitor **Done**
  - Data Loader Optimization **Working on**
- Experiments



데이터셋 확보, 재구현, 모델 매니져 및 모니터링 구현 완료
데이터 로더 구현 완료 후 최적화 과정 진행중

Current Status: Dataset is big

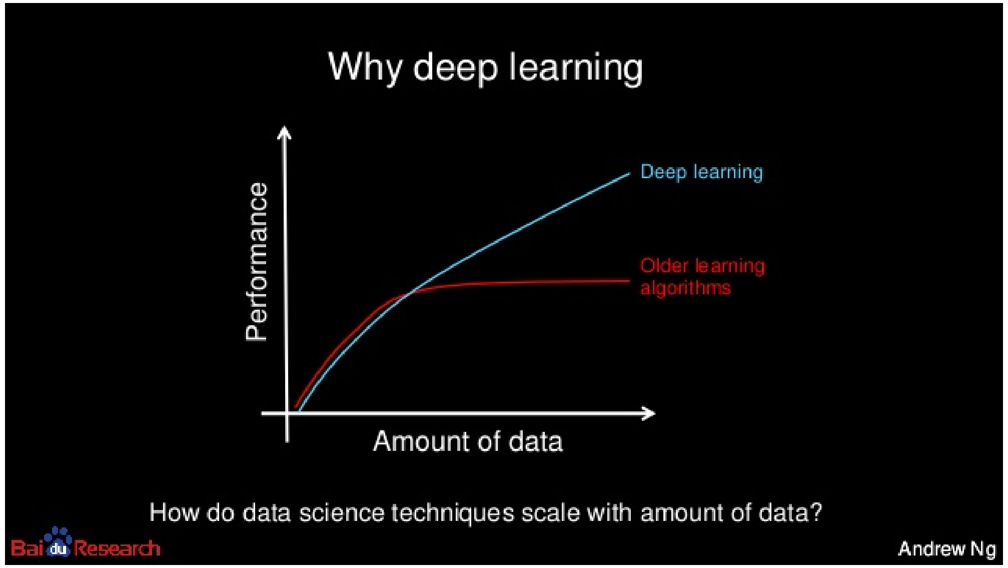
- Does not fit on disk on VM
- Dataset spp: 128, 256, 512, 1024, 8192
- 1451 images per spp
- Total size of images: 294 Gb
- Sampling 400 patches per image: 300 Mb
    - Importance sampling reduces to 150 Mb
- At least 128 and 8192 spp required

→ 1451 * 2 * 150 Mb = 435 Gb

Of storage needed for just patches

→ 1451 * 5 * 150 Mb + 294 Gb = 1382 Gb

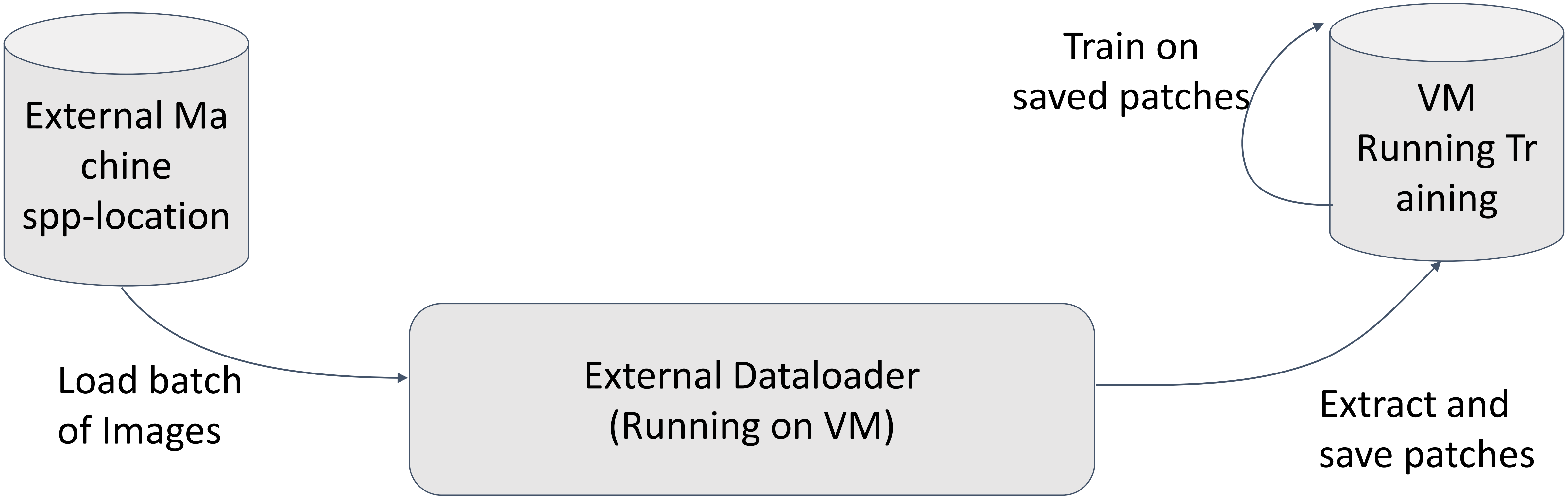
To save all data

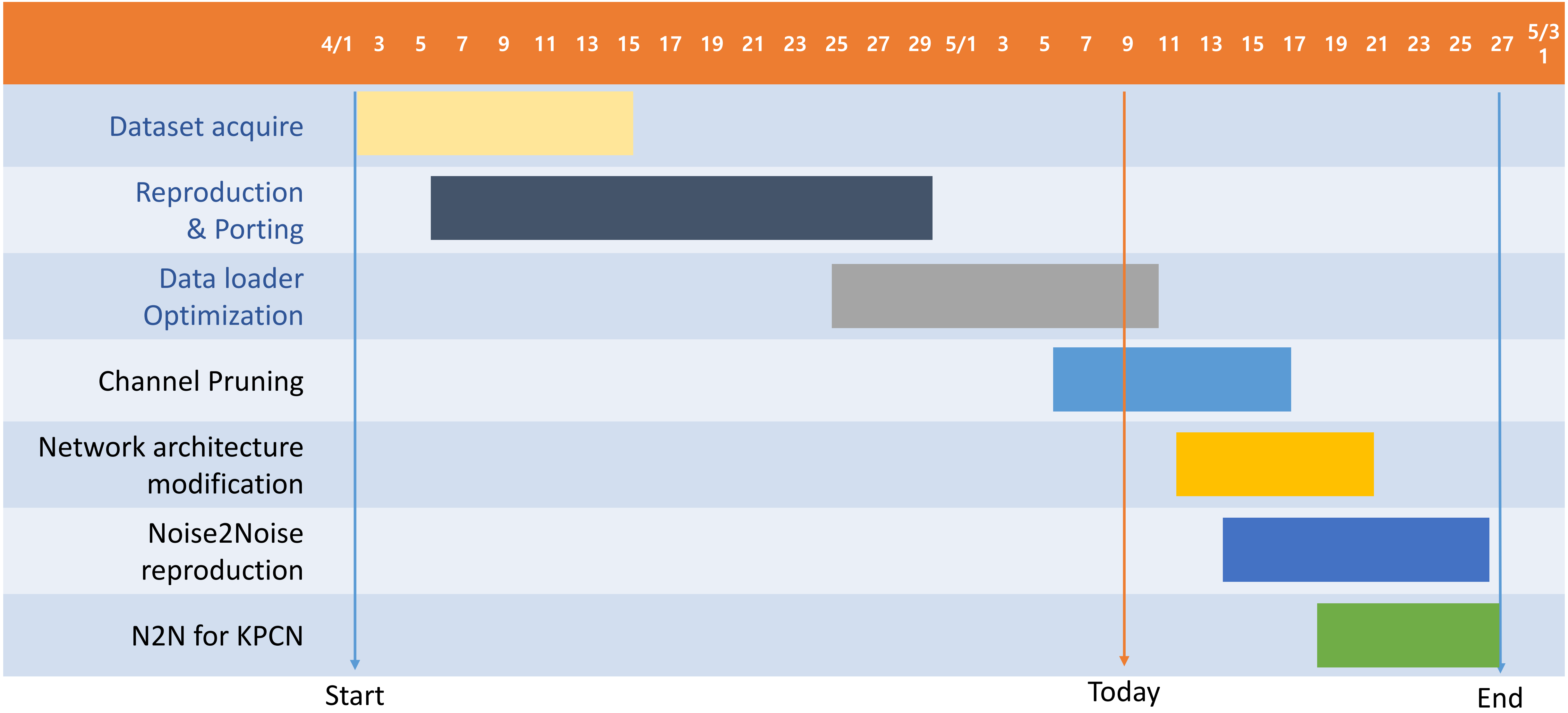


https://www.slideshare.net/ExtractConf/andrew-ng-chief-scientist-at-baidu/30

Solution: External Data Loader

- Loading patch < Loading image + patch sampling
- Outsource data storage
- Requires enough storage to save all patches
- WIP




4/1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 5/1 3 5 7 9 11 13 15 17 19 21 23 25 27 5/31
Dataset acquire
Reproduction & Porting
Data loader Optimization
Channel Pruning
Network architecture modification
Noise2Noise reproduction
N2N for KPCN
Start
Today
End

Team Contribution

- Cheolmin Lee: Reproduction & Baseline-Code implementation
- Minki Jo: Theoretical Approach/Baseline-Code implementation
- Nick Heppert: Theoretical Approach/External Data Loader

Thank You!

Reference

[Liu et. al. 17] Learning Efficient Convolutional Networks through Network Slimming, ICCV2017

[Zhang et. al. 18] Image Super-Resolution Using Very Deep Residual Channel Attention Networks, ECCV2018

[Lehtinen et. al. 18] Noise2Noise: Learning Image Restoration without Clean Data, ICML2018

[Bako Et al. 17] "Kernel-Predicting Convolutional Networks for Denoising Monte Carlo Renderings." ACM Transactions on Graphics 36, no. 4 (July 20, 2017)