

삼성 오픈소스 컨퍼런스

태양에서 세포까지 인공지능 활용 사례

(주)인스페이스 | 미래기술실 | 김태영

기술이사

2018-10-17

SOSCON 2018

SAMSUNG OPEN SOURCE CONFERENCE 2018



You are in **Intelligence Space**.



Sun to Cell



```
import autokeras as ak
```

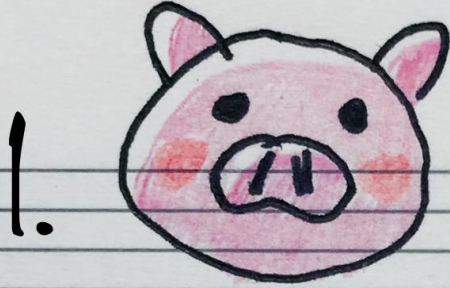
```
clf = ak.ImageClassifier()
```

```
clf.fit(x_train, y_train)
```

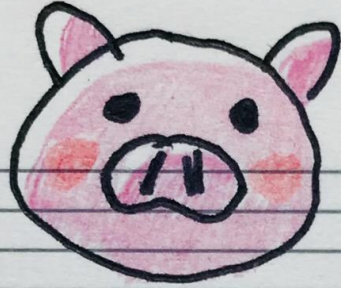
```
results = clf.predict(x_test)
```

문제 정의와 여러 풀이법

Which of the following is not a mammal?

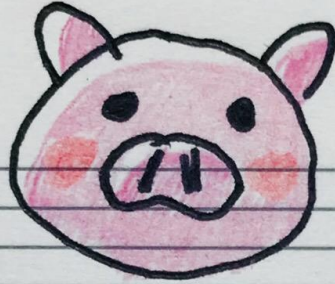


Which of the following is not a mammal?

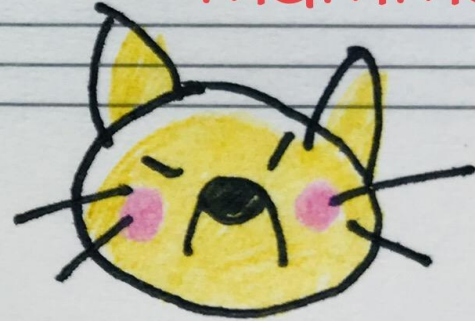


Which of the following is not a mammal?

mammal



mammal

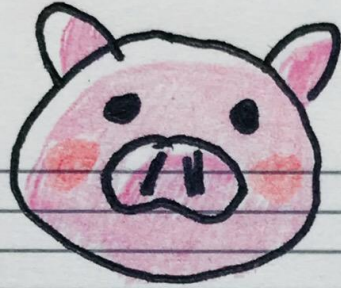


mammal



reptile

Which of the following is not a mammal?



-10



-10



-10



+10



인공지능(딥러닝)의 발전

딥러닝 이야기

빅데이터와 딥러닝

케라스 이야기

케라스 개념잡기

모델 개념잡기

태양에서 세 포까지

극한알바

파이썬 기초

레이어 개념잡기

딥브릭 이야기

생성모델(GAN) 살펴보기

딥강화학습 살펴보기

딥러닝 최신동향 살펴보기

딥러닝 이야기

Lec 00 - Machine/Deep learning 수업의 개요와 일정

조회수 191,707회



Sung Kim

게시일: 2016. 3. 21.

$$Y = w \times X + b$$

“이 정도의 수학을 이해할 수 있으면 충분히 따라올 수가 있고 머신러닝을 이해할 수가 있습니다” 김성훈 교수님

실측값

$$Y = w \times X + b$$

예측값

$$\hat{Y} = w \times \hat{X} + b$$

실측값

$$Y = w \times X + b$$



예측값

$$\hat{Y} = w \times \hat{X} + b$$

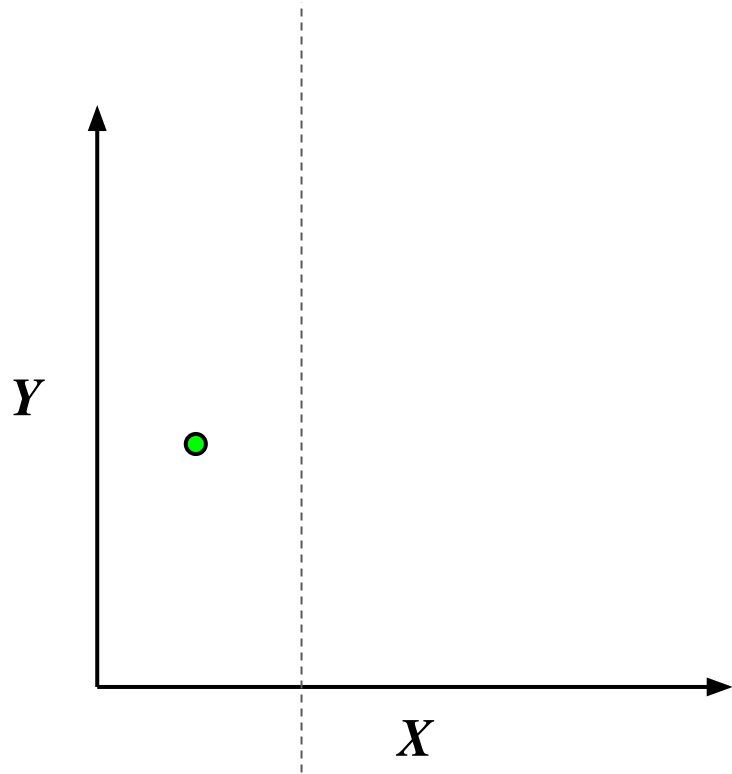
w 와 b 를 구해봅시다.

실측값

$$3 = w \times 1 + b$$

예측값

$$\hat{Y} = w \times 2 + b$$



w 와 b 를 구해봅시다.

실측값

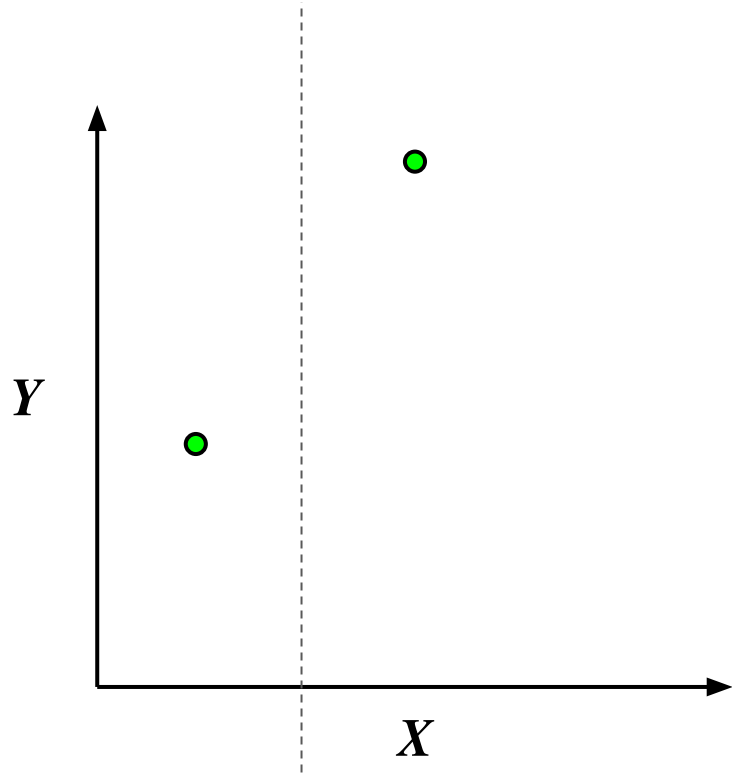
$$3 = w \times 1 + b$$

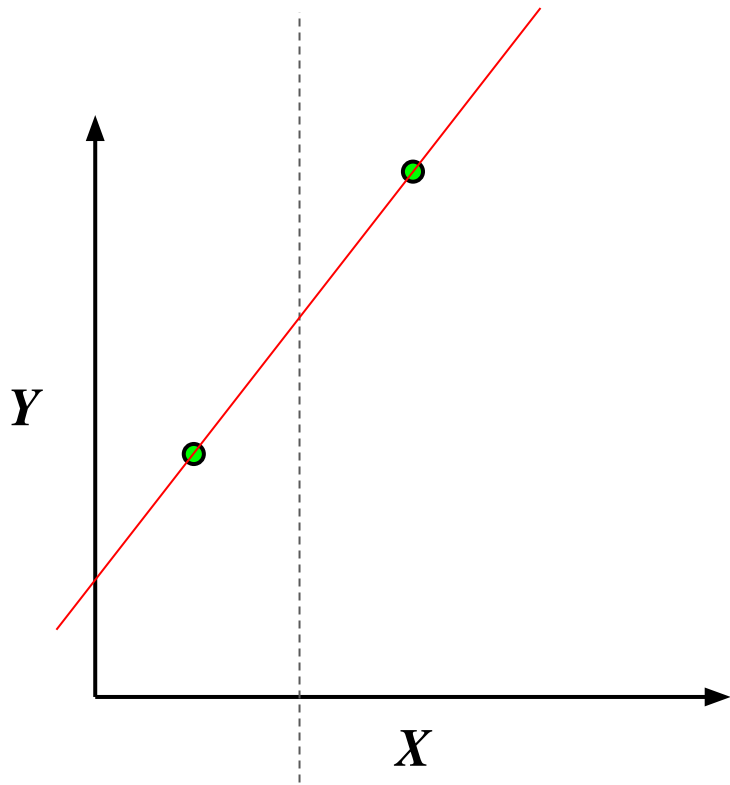
$$5 = w \times 3 + b$$



예측값

$$\hat{Y} = w \times 2 + b$$





w 와 b 를 구해봅시다.

실측값

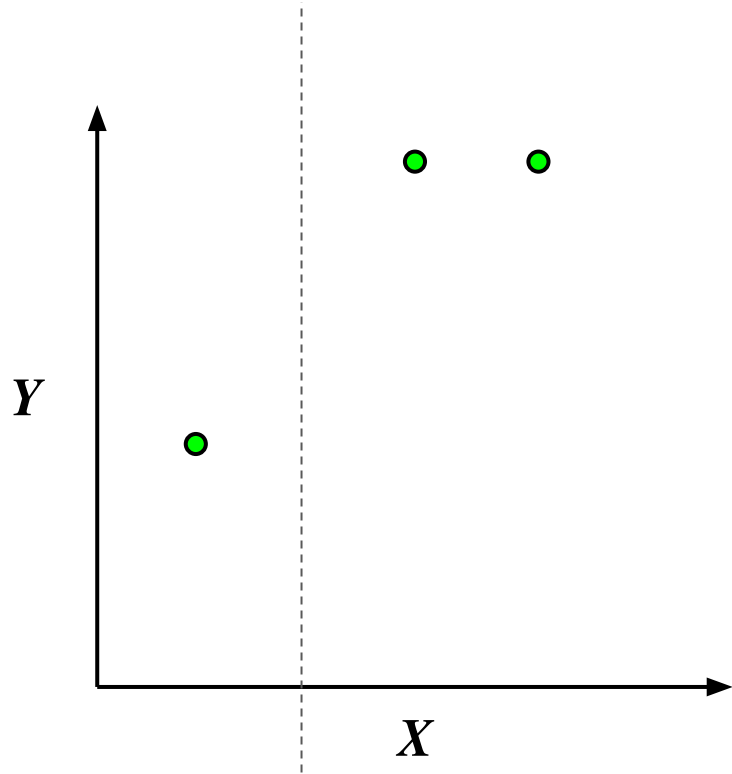
$$3 = w \times 1 + b$$

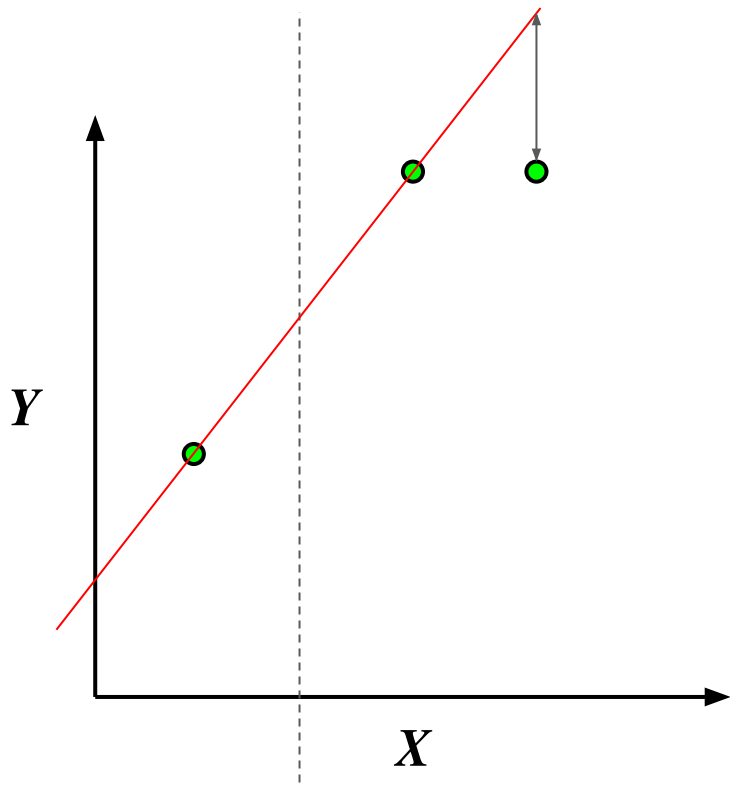
$$5 = w \times 3 + b$$

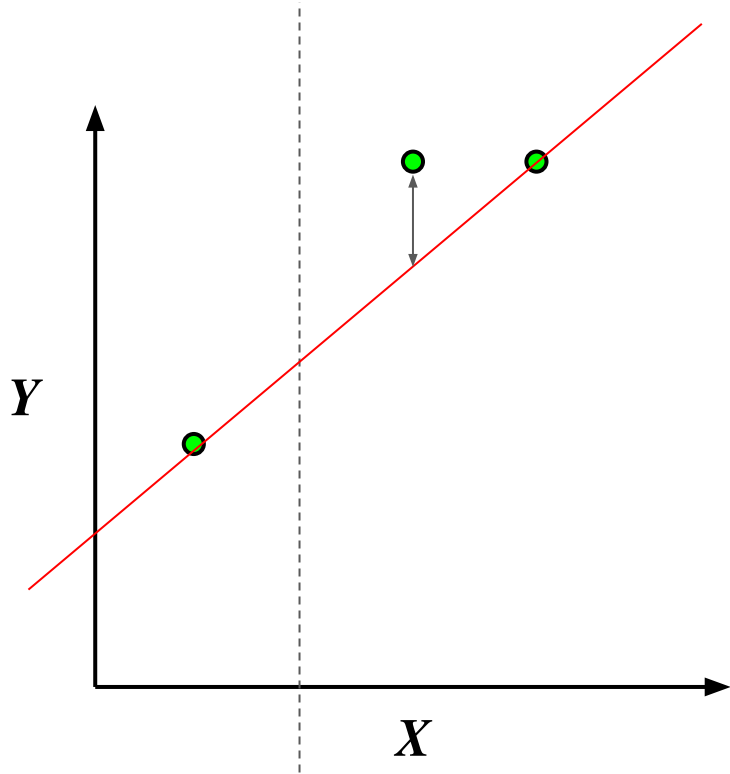
$$5 = w \times 4 + b$$

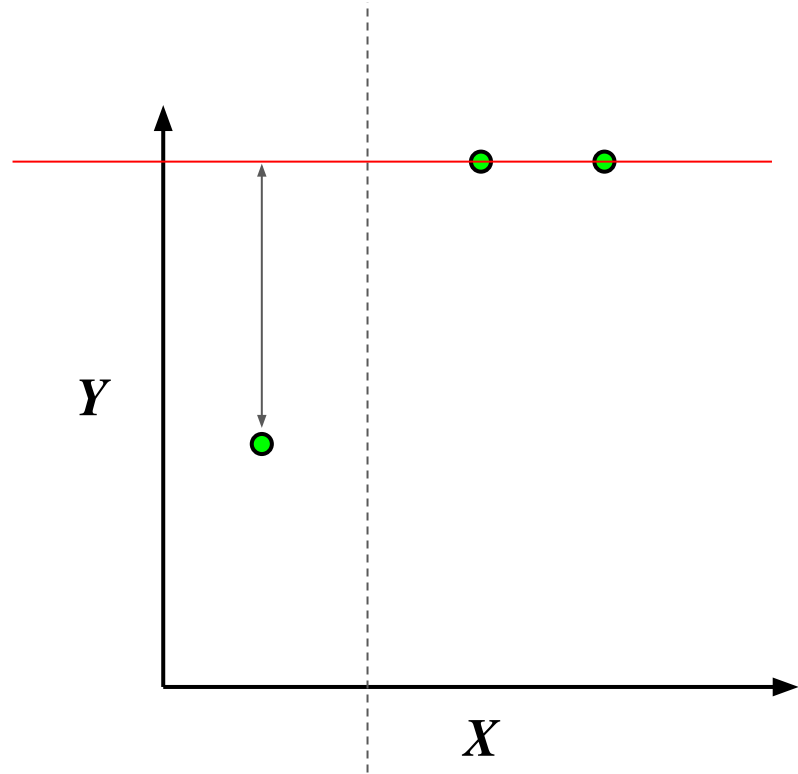
예측값

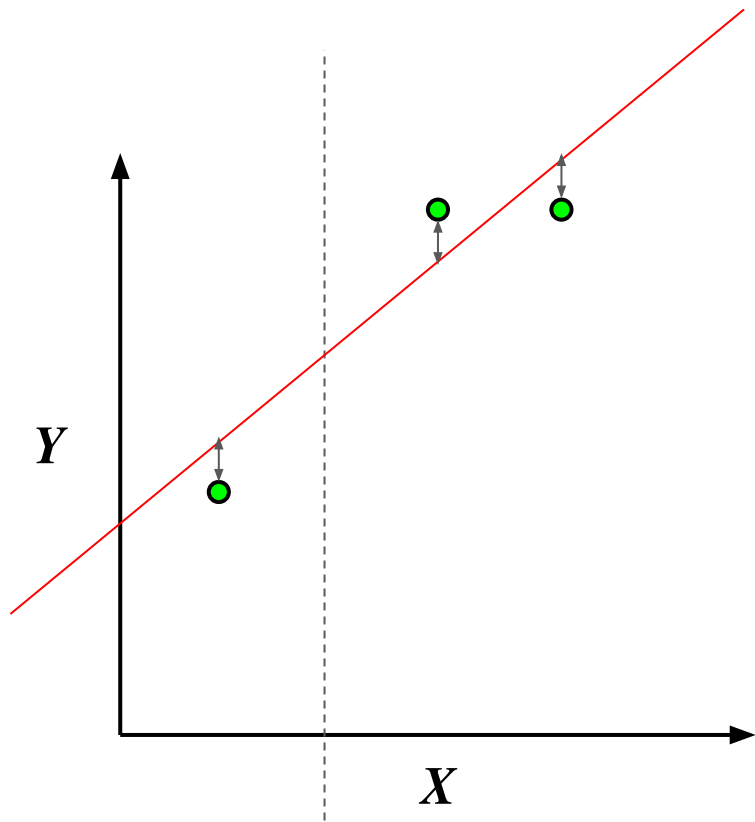
$$\hat{Y} = w \times 2 + b$$











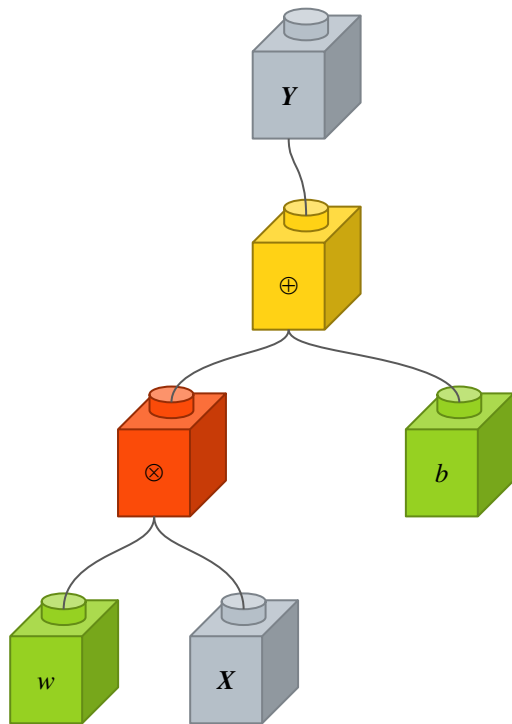
우리가 w 와 b 를 구해봅시다.

$$Y = w \times X + b$$

$$3 = w \times 1 + b$$

$$5 = w \times 3 + b$$

$$5 = w \times 4 + b$$



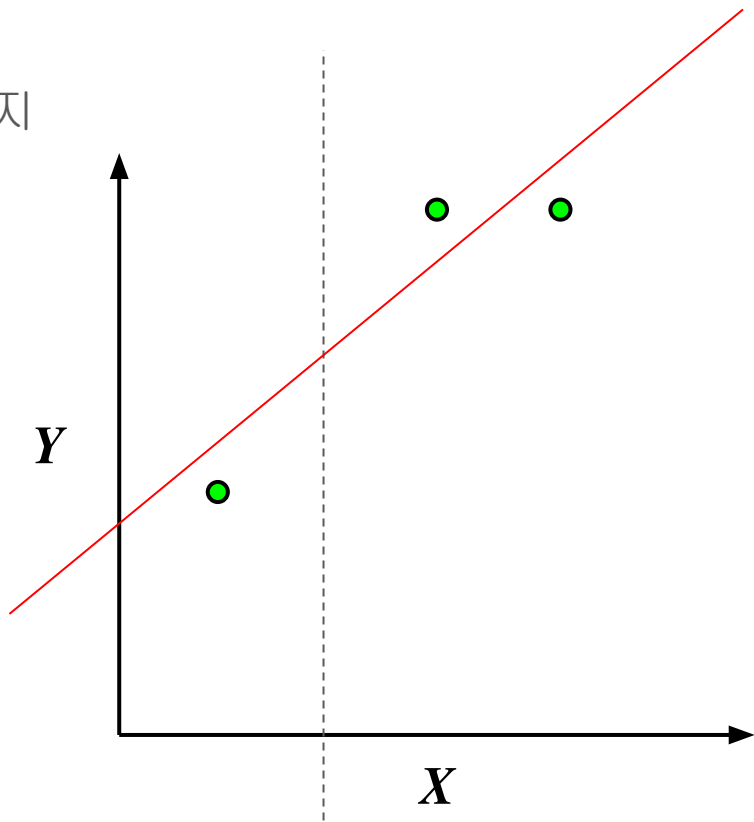
수식으로 나이스하게 풀던 지

$$\begin{aligned}
 na_0 + \sum x_i a_1 - \sum y_i &= 0 \\
 \sum x_i a_0 + \sum x_i^2 a_1 - \sum x_i y_i &= 0 \\
 a_0 + \frac{\sum x_i}{n} a_1 - \frac{\sum y_i}{n} &= 0 \\
 a_0 + \frac{\sum x_i^2}{\sum x_i} a_1 - \frac{\sum x_i y_i}{\sum x_i} &= 0 \\
 \left(\frac{\sum x_i}{n} - \frac{\sum x_i^2}{\sum x_i} \right) a_1 &= \frac{\sum y_i}{n} - \frac{\sum x_i y_i}{\sum x_i} \\
 a_1 &= \frac{\frac{\sum y_i}{n} + \frac{\sum x_i y_i}{\sum x_i} \times \frac{n \sum x_i}{\sum x_i^2 - m \sum x_i^2}}{\frac{\sum x_i}{n} - \frac{\sum x_i^2}{\sum x_i}} \\
 &= \frac{\sum x_i \sum y_i + m (\sum x_i y_i)}{(\sum x_i)^2 - m \sum x_i^2} \\
 m \sum x_i^2 a_0 + \sum x_i \sum x_i^2 a_1 - \sum x_i^2 \sum y_i &= 0 \\
 (\sum x_i)^2 a_0 + \sum x_i \sum x_i^2 a_1 - 2 \sum x_i \sum x_i y_i &= 0 \\
 (m \sum x_i^2 - (\sum x_i)^2) a_0 &= \frac{\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i}{\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i} \\
 \therefore a_0 &= \frac{m \sum x_i^2 - (\sum x_i)^2}{\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i}
 \end{aligned}$$

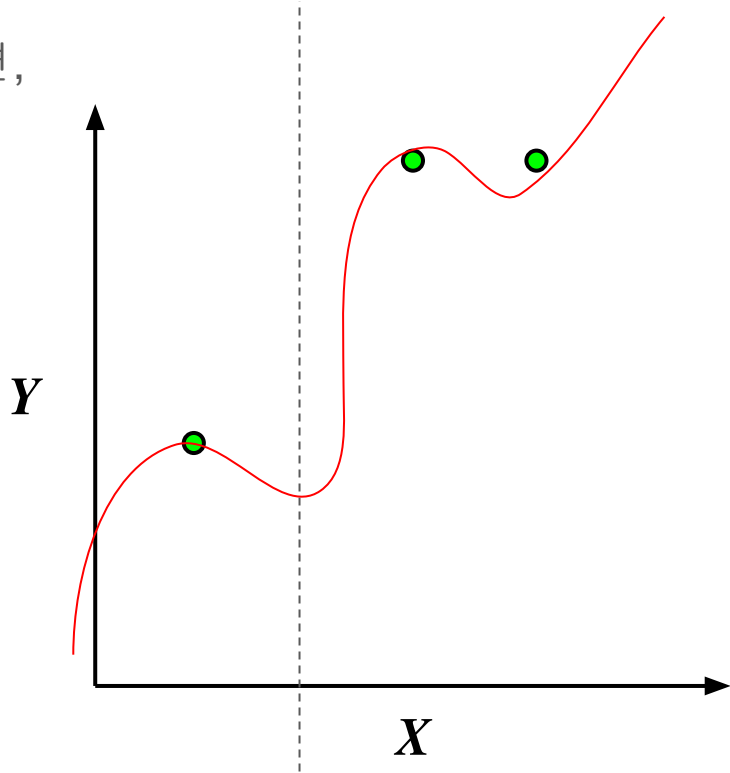
계산할때 $\sum x_i$ $\sum x_i^2$ $\sum y_i$ $\sum x_i y_i$

$$\begin{aligned}
 a_1 &= \frac{\frac{\sum x_i y_i}{\sum x_i} - \bar{y}}{\frac{\sum x_i^2}{\sum x_i} - \bar{x}} & a_0 &= \bar{y} - a_1 \bar{x} \\
 0 \sum x_i a_0 + \sum x_i^2 a_1 - \sum x_i y_i &= 0 \\
 \sum x_i \bar{y} - a_1 \bar{x} \sum x_i + \sum x_i^2 a_1 - \sum x_i y_i &= 0 \\
 \sum x_i^2 - \sum x_i \bar{x} & a_1 (\sum x_i^2 - \sum x_i \bar{x}) = \sum x_i y_i - \sum x_i \bar{y} \\
 = \sum x_i^2 - 2 \sum x_i \cdot \frac{\sum x_i}{n} & a_1 = \frac{\sum x_i y_i - \sum x_i \bar{y}}{\sum x_i^2 - \sum x_i \bar{x}} \\
 = \sum x_i^2 - \frac{(\sum x_i)^2}{n} & \\
 = \sum x_i^2 - 2 \cdot \frac{(\sum x_i)^2}{n} + \frac{(\sum x_i)^2}{n} & = \sum x_i y_i - \sum x_i \bar{y} \\
 = \sum x_i^2 - 2 \sum x_i \cdot \frac{\sum x_i}{n} + \left(\frac{\sum x_i}{n} \right)^2 \cdot n & \rightarrow \sum (x_i - \bar{x})(y_i - \bar{y}) \\
 = \sum x_i^2 - 2 \sum x_i \bar{x} + m (\bar{x})^2 & = \sum x_i y_i - \sum x_i \bar{y} \\
 \sum x_i - 2 \sum x_i \bar{x} + \bar{x}^2 + \bar{x} & = \sum x_i y_i - \sum x_i \bar{y} \\
 = x_1 - 2 \sum x_1 \bar{x} + \bar{x}^2 + \dots x_n - 2 \sum x_n \bar{x} + \bar{x}^2 & = \sum x_i y_i - \sum x_i \bar{y} \\
 = \sum (x_i - \bar{x})^2 & \text{Var}(x) \\
 \sum (x_i - \bar{x})(y_i - \bar{y}) & \sum x_i y_i - \sum x_i \bar{y} \\
 \sum (x_i y_i - \bar{x} y_i - \bar{y} x_i + \bar{x} \bar{y}) & = \sum x_i y_i - 2 \frac{\sum x_i \sum y_i}{n} + \frac{2 \sum x_i \sum y_i}{n \cdot n} \\
 = \sum x_i y_i - \bar{x} \sum y_i - \bar{y} \sum x_i + m \bar{x} \bar{y} & = \sum x_i y_i - \bar{x} \sum y_i - \bar{y} \sum x_i + m \bar{x} \bar{y} \\
 공분산 covariance & = \sum (x_i y_i - \bar{x} y_i - \bar{y} x_i + \bar{x} \bar{y}) \\
 2개의 상관정도를 나타내는 것 & = \sum x_i y_i - \bar{x} \sum y_i - \bar{y} \sum x_i + m \bar{x} \bar{y} \\
 2개 변수를 하나가 줄어질때, 다른 하나도 줄어드는 것. & \\
 공분산이 0 이면 두개가 독립변수일 때. &
 \end{aligned}$$

또는 모두 넣어보던지

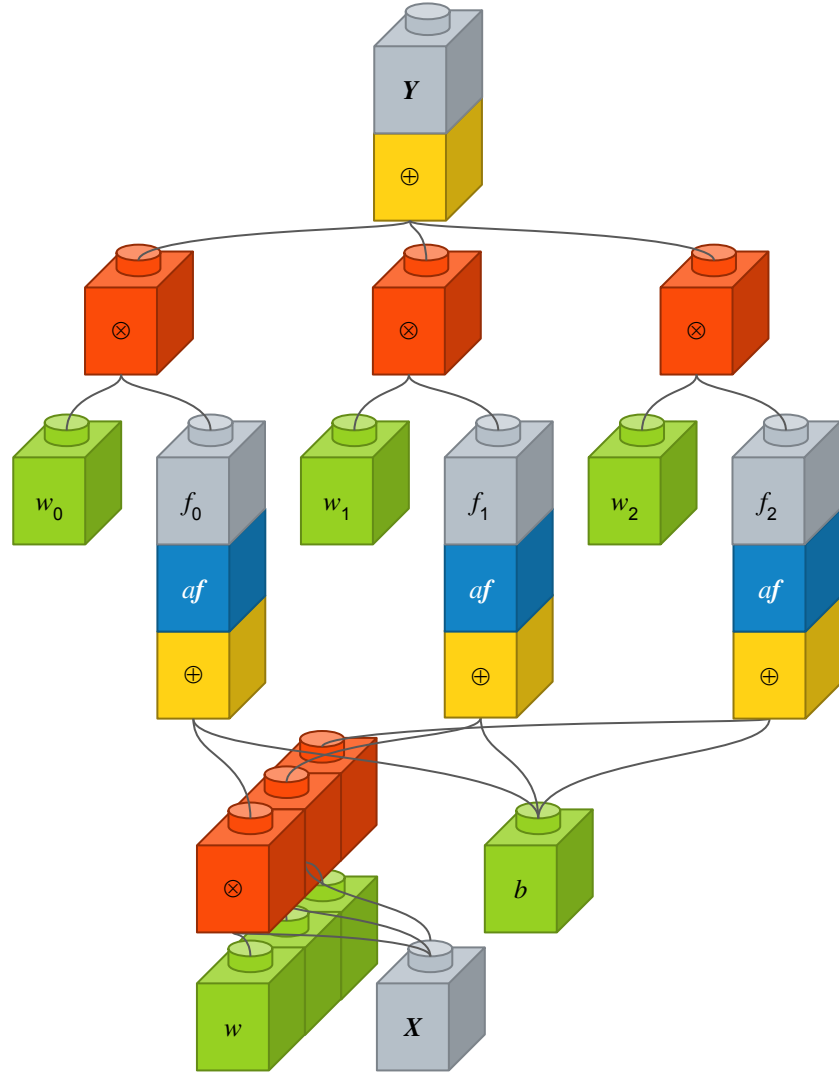


만약 이런 모델이라면,



w 와 b 는 딥러닝 모델이 알아서
구해줍니다.

$$Y = w \times X + b$$

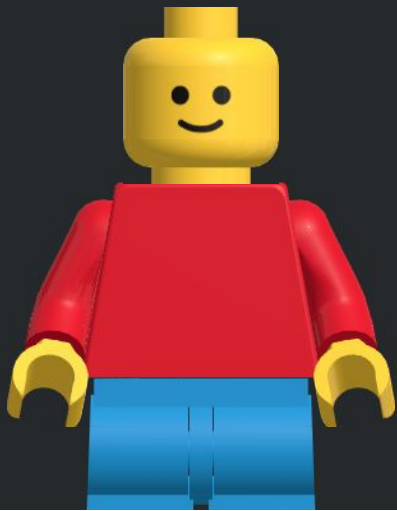


모델 개념잡기



Model

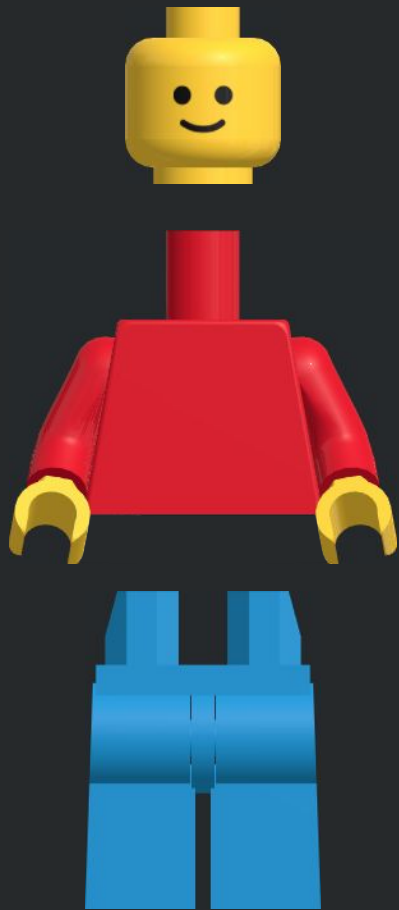
모델

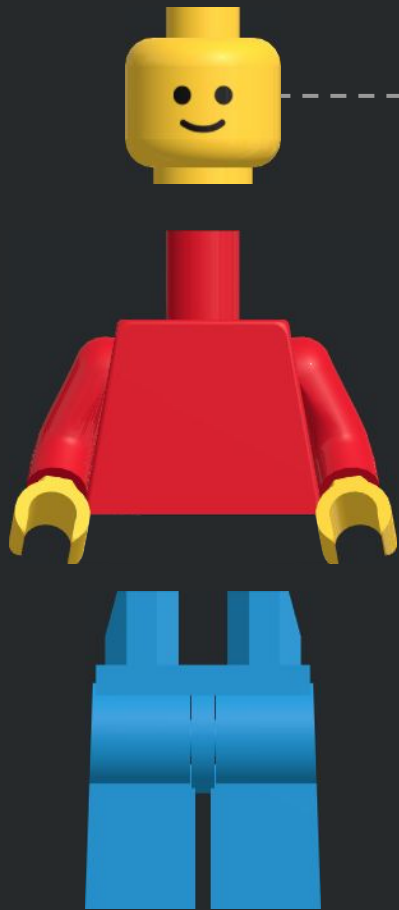


Model

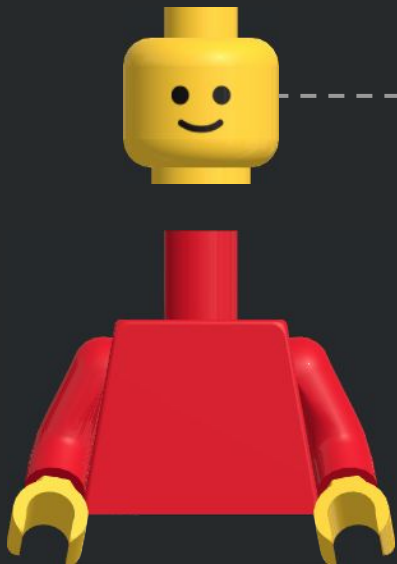
모델

```
from keras.models import Sequential  
  
model = Sequential()
```



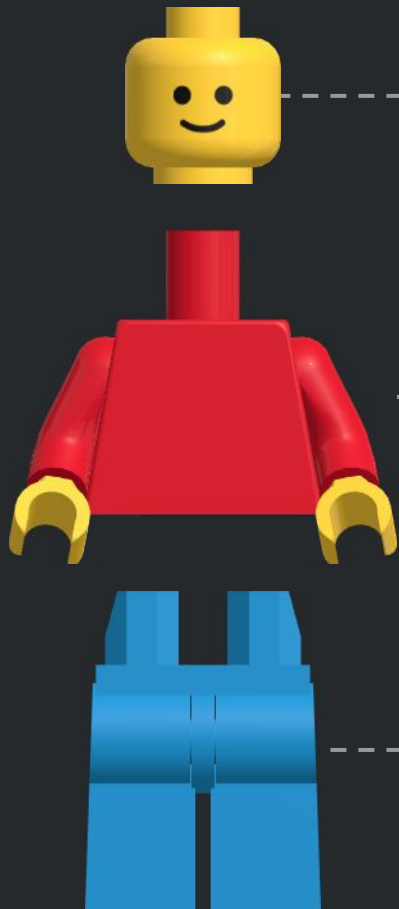


Network
네트워크



Network
네트워크

```
from keras.layers import Dense  
  
model.add(Dense(units=64, activation='relu', input_dim=100))  
model.add(Dense(units=10, activation='softmax'))
```

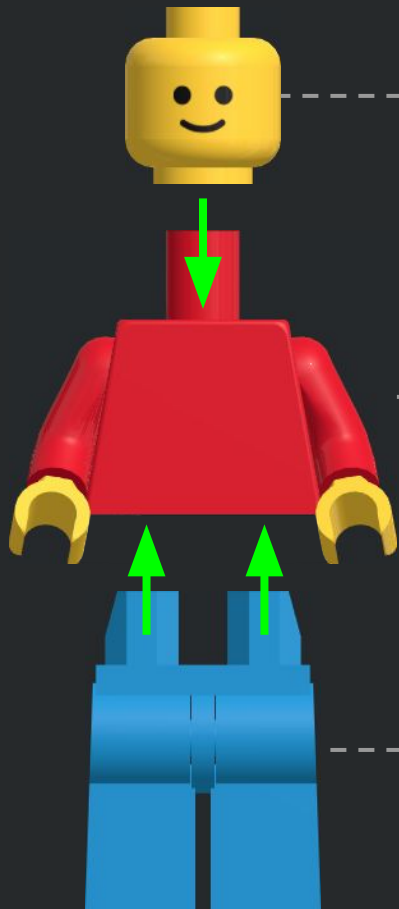


Network
네트워크

Objective Function
목표함수

Optimizer
최적화기

Compile
컴파일



Network
네트워크

Objective Function
목표함수

Optimizer
최적화기

Compile
컴파일



Network
네트워크

Objective Function
목표함수

```
model.compile(loss='categorical_crossentropy',  
              optimizer='sgd',  
              metrics=['accuracy'])
```


Model

모델



네트워크

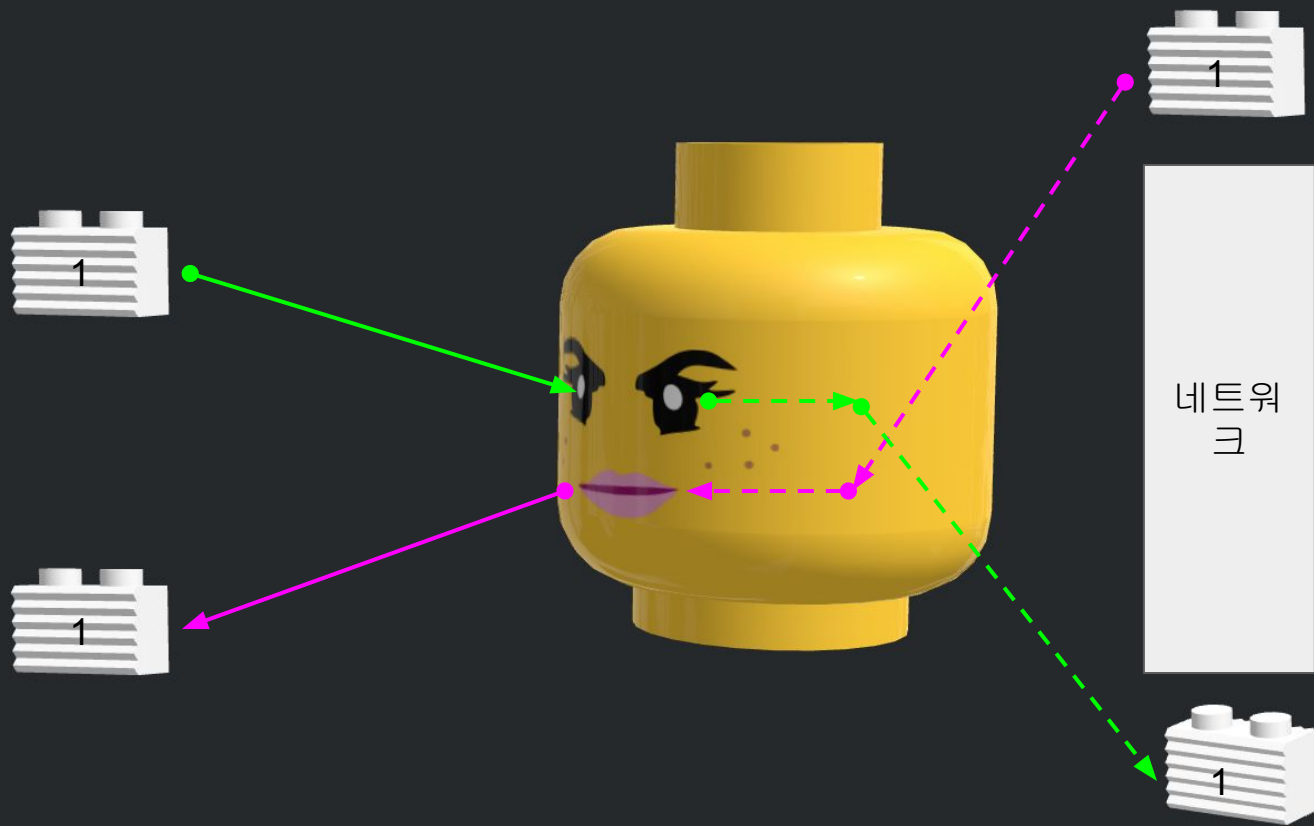
network

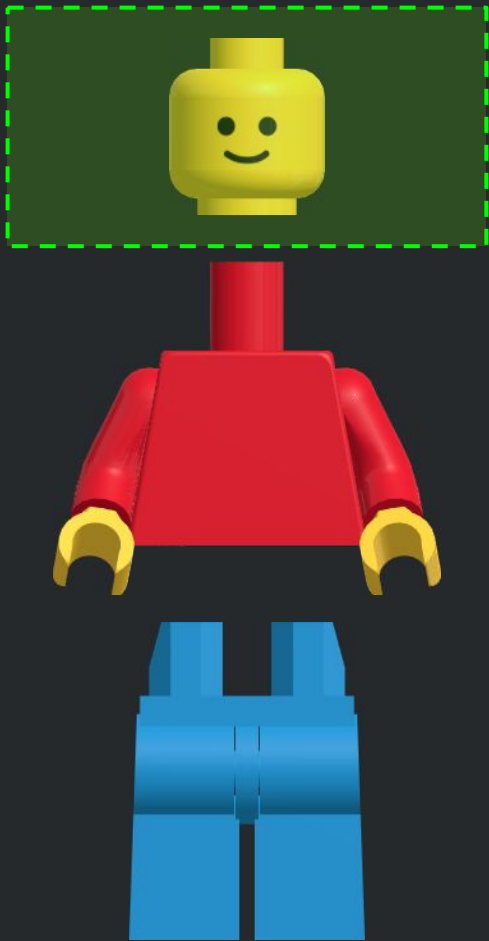
목표함수

objective function

최적화기

optimizer

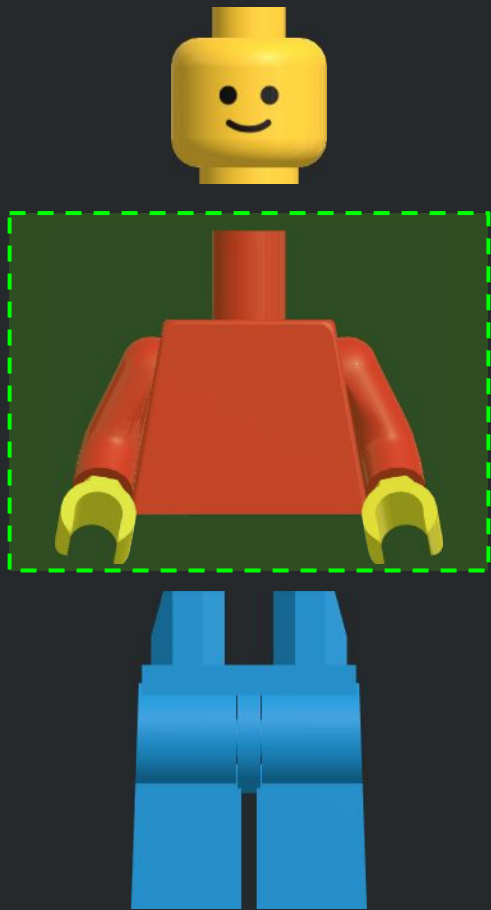




<https://keras.io/applications/>

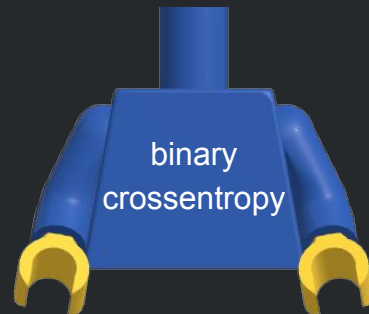
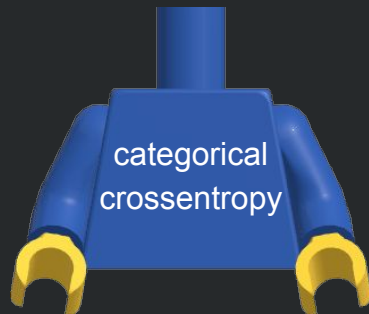
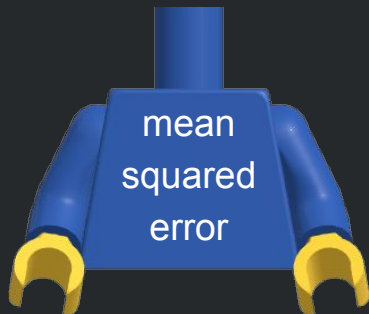
- Xception (88 MB, 126)
- VGG16 (528 MB, 23)
- VGG19 (549 MB, 26)
- ResNet50 (99 MB, 168)
- InceptionV3 (92 MB, 159)
- InceptionResNetV2 (215 MB, 572)
- MobileNet (17 MB, 88)
- DenseNet121 (33 MB, 121)
- DenseNet169 (57 MB, 169)
- DenseNet201 (80 MB, 201)
- NASNet

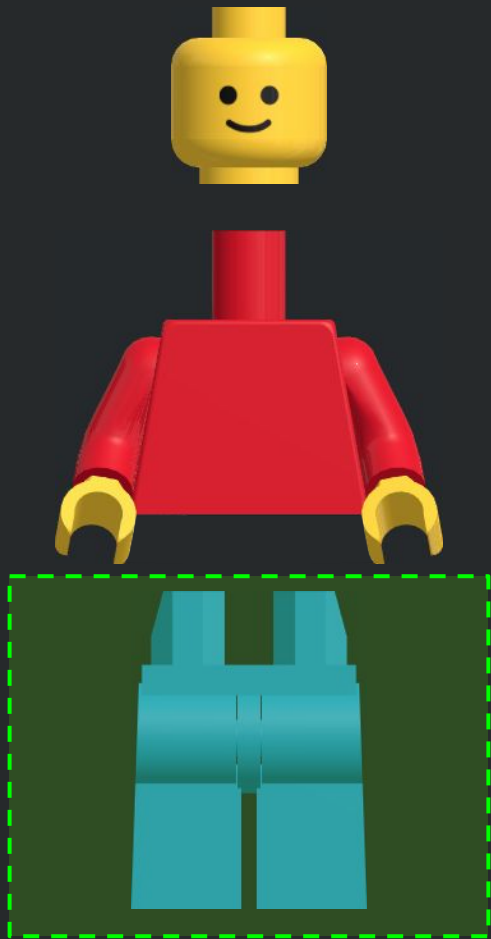




<https://keras.io/losses/>

- **mean_squared_error**
- mean_absolute_error
- mean_absolute_percentage_error
- mean_squared_logarithmic_error
- squared_hinge
- hinge
- categorical_hinge
- logcosh
- **categorical_crossentropy**
- sparse_categorical_crossentropy
- **binary_crossentropy**
- kullback_leibler_divergence
- poisson
- cosine_proximity





<https://keras.io/optimizers/>

- SGD
- RMSprop
- Adagrad
- Adadelata
- Adam
- Adamax
- Nadam
- TFOptimizer



SGD



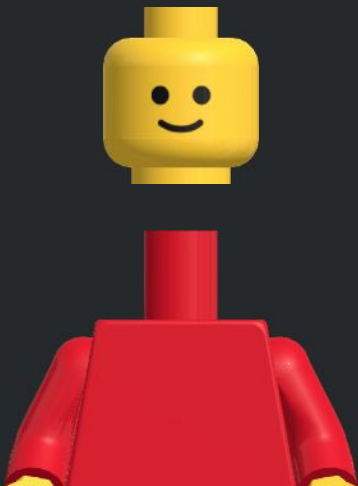
RMSprop



Adagrad



Adam



<https://keras.io/optimizers/>

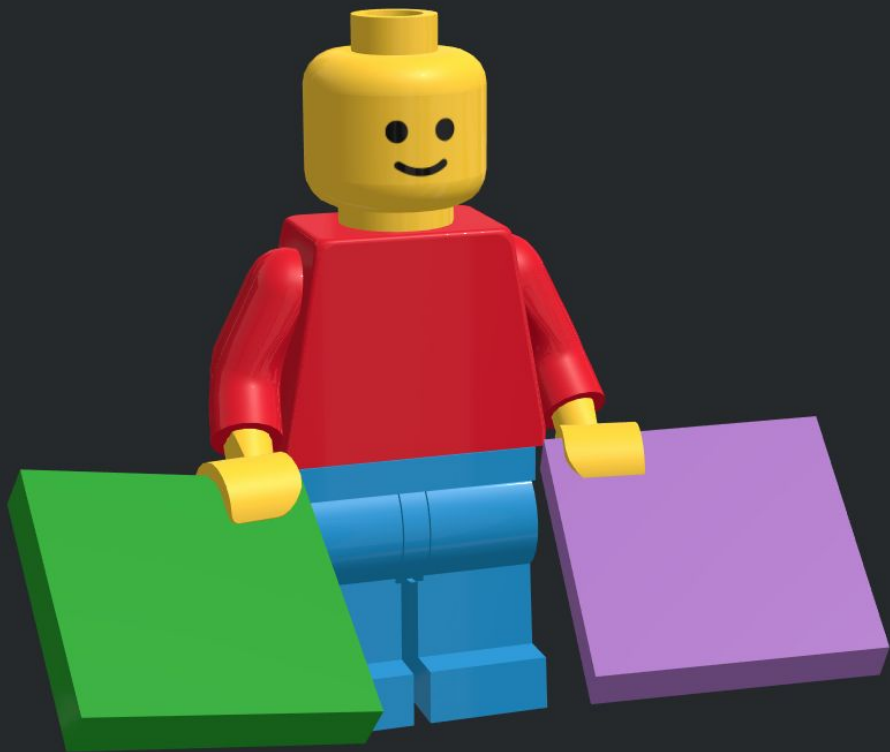
- SGD
- RMSprop
- Adagrad
- Adadelata
- Adam
- Adamax

```
sgd = optimizers.SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='mean_squared_error', optimizer=sgd)
```

```
# pass optimizer by name: default parameters will be used
model.compile(loss='mean_squared_error', optimizer='sgd')
```

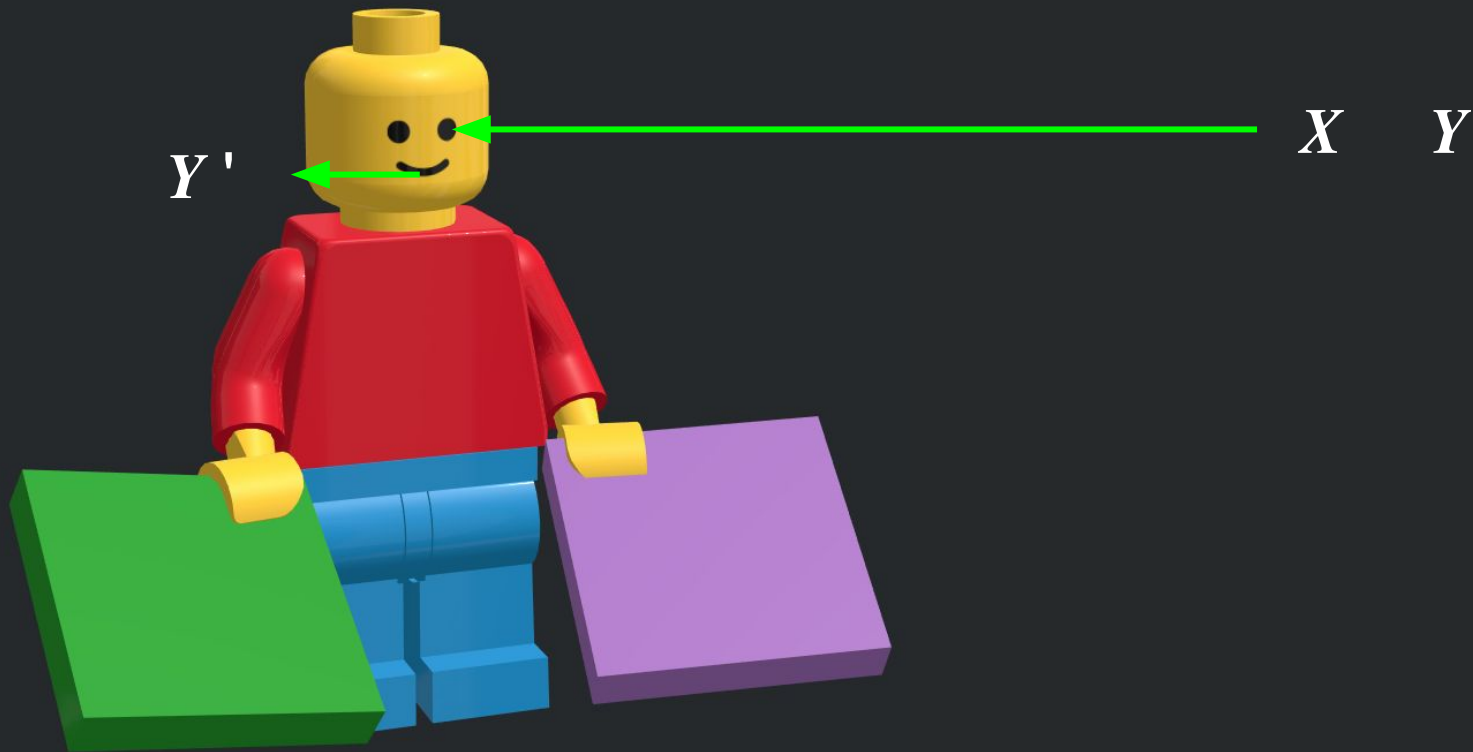




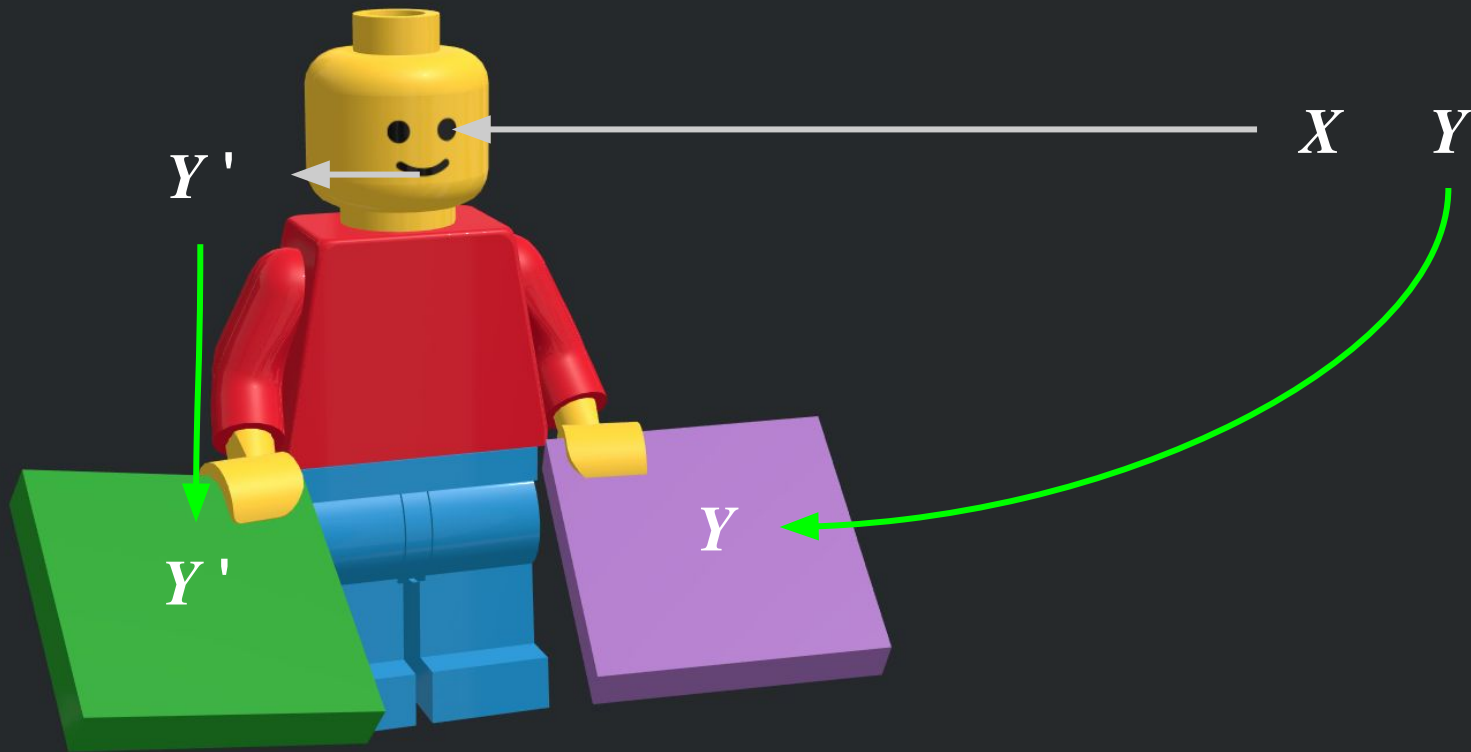


X Y

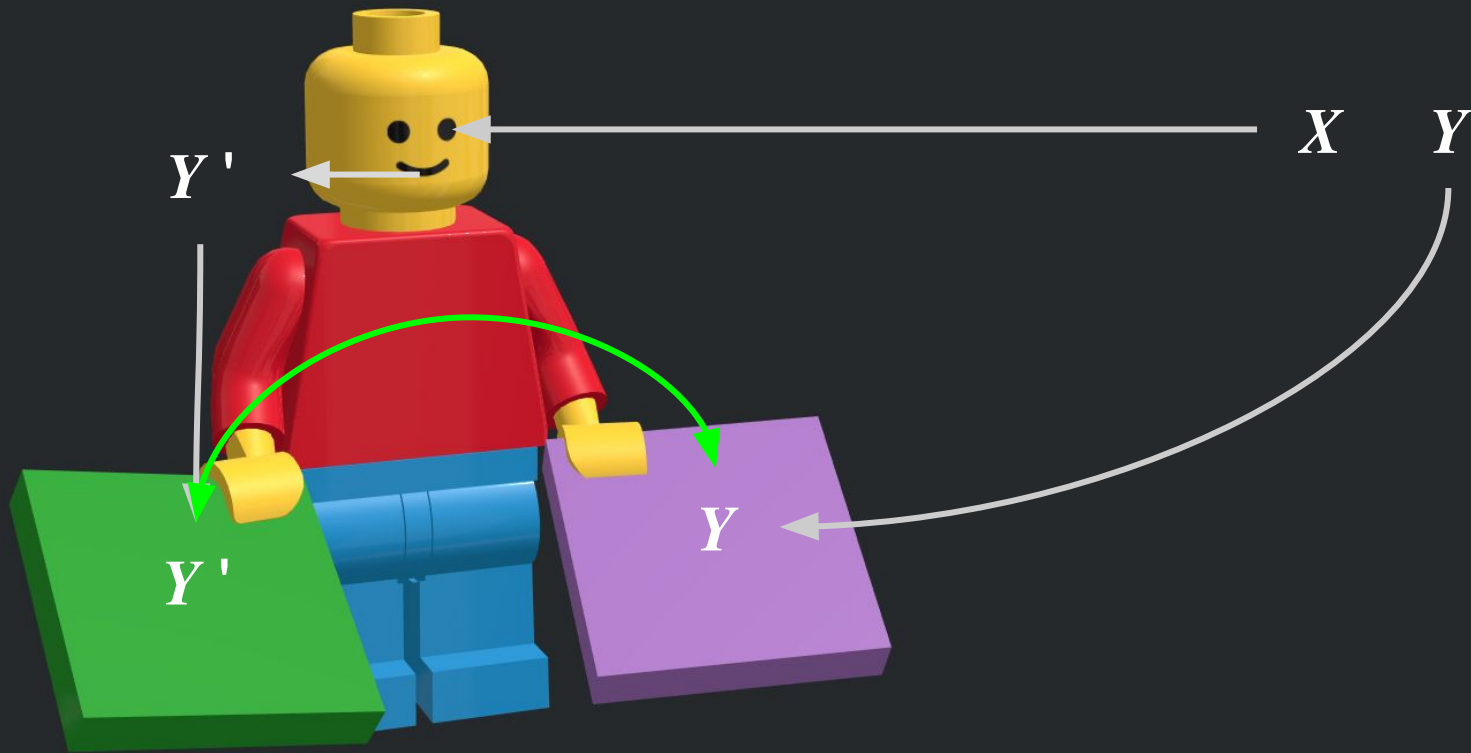
$$Y' = w \times X + b$$



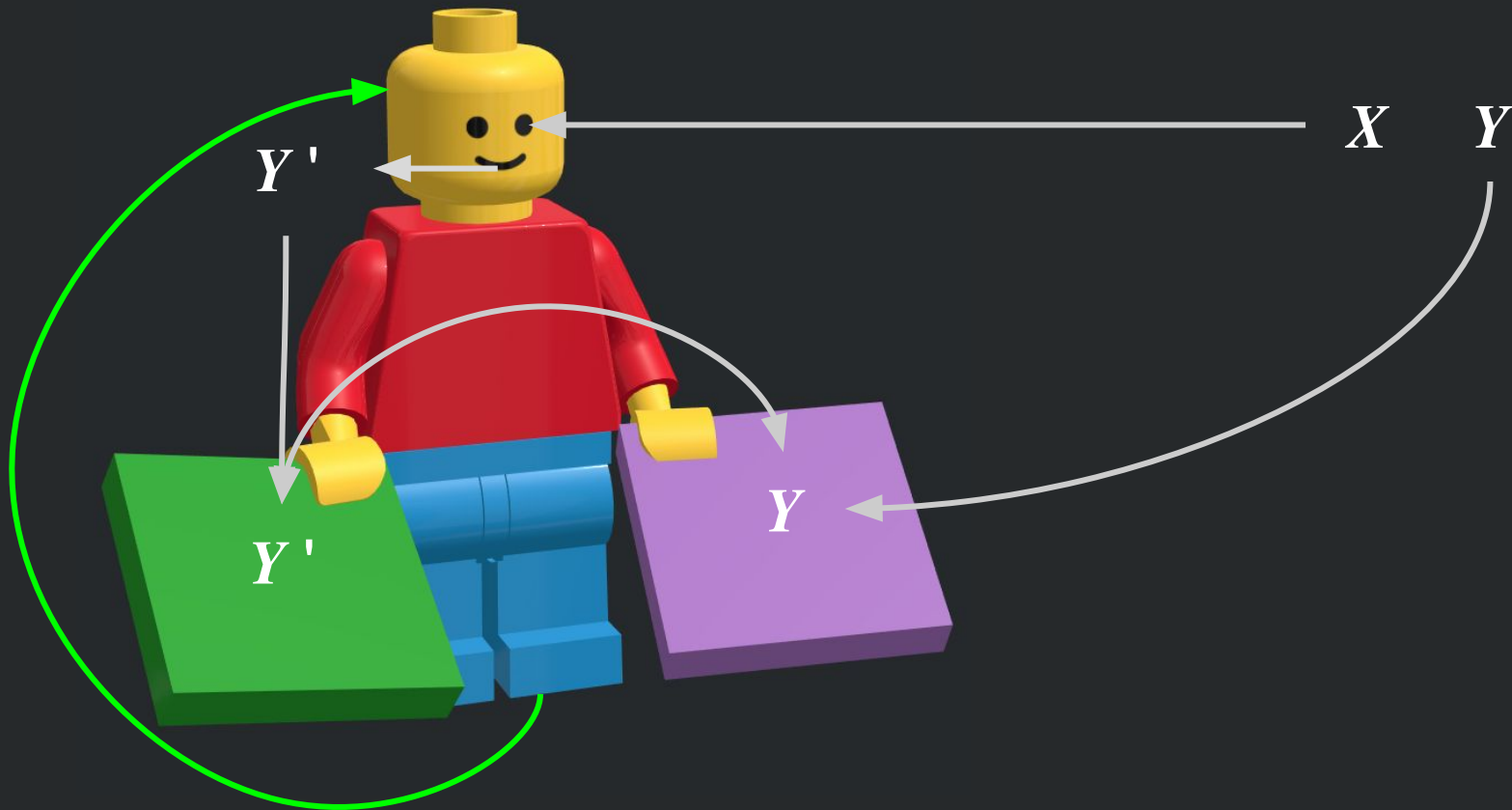
$$Y' = w \times X + b$$



$$Y' = w \times X + b$$



$$Y' = w \times X + b$$



태양에서 세포까지 극한알바

You are in **Intelligence Space**.



Sun to Cell



딥러닝으로 어떤 문제를 풀까?

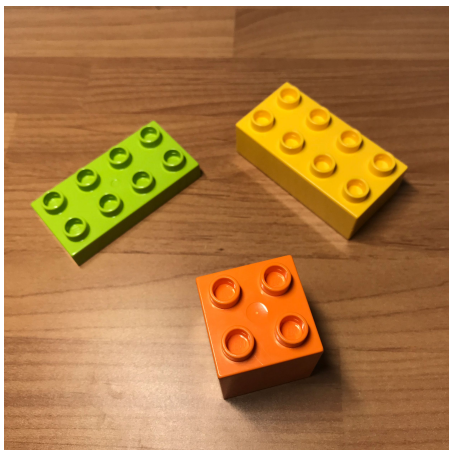
기존 알고리즘을 교체
딥러닝으로 풀기 가능해진 문제들
새로운 문제 정의

기존 알고리즘을 교체

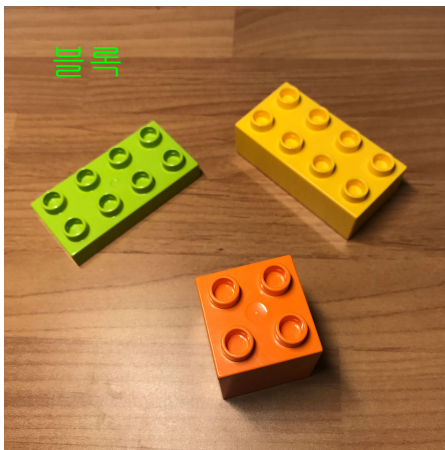
딥러닝으로 풀기 가능해진 문제들

새로운 문제 정의

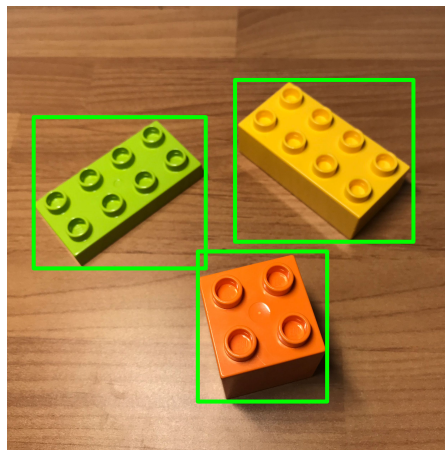
사진



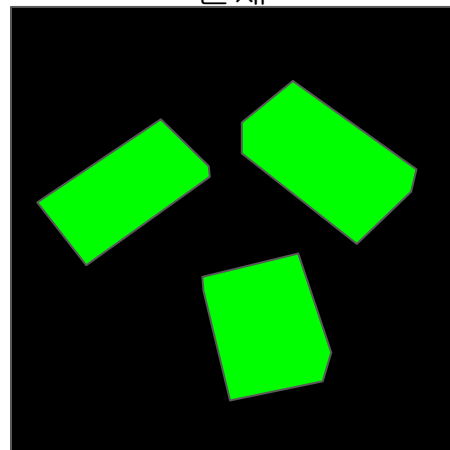
분류 문제



객체 검출 문제



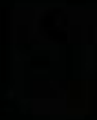
의미론적 분할
문제



기존 알고리즘을 교체

딥러닝으로 풀기 가능해진 문제들

새로운 문제 정의



Conference on Computer Vision and Pattern Recognition



17 June - 2 July 2004

Las Vegas, NV

IEEE Computer Society

IEEE Computer Society
Conference on
Computer Vision and
Pattern Recognition

27 June - 2 July 2004

WASHINGTON, D.C.

Sponsored by the IEEE Computer Society

IEEE Computer Society Conference on
Computer Vision and Pattern Recognition
Volume II

EIGHTH INTERNATIONAL CONFERENCE
ON COMPUTER VISION
VOLUME II



IEEE ICRA-2004

•

Performance Digest

• April 26-May 1, 2004, New Orleans, LA

2004 CVPR Abstract Book

2004 CVPR Abstract Book

IEEE

IEEE



IEEE Computer Society
Conference on
Computer Vision and
Pattern Recognition

27 June - 2 July 2004
WASHINGTON, D.C.
Sponsored by the IEEE Computer Society

IEEE Computer Society Conference on
Computer Vision and Pattern Recognition
Volume II



EIGHTH INTERNATIONAL CONFERENCE
ON COMPUTER VISION
VOLUME II



IEEE CVPR 2004 • Conference Digest • April 26-May 1, 2004, Washington, DC • IEEE

2004 CVPR Abstract Book

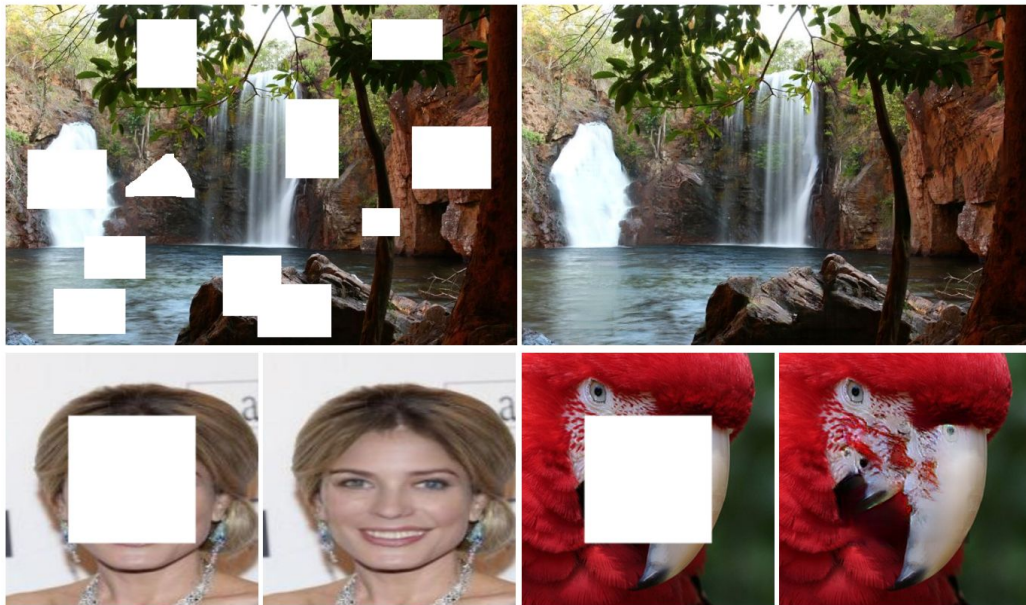
2004 CVPR Tutorial Book

Generative Image Inpainting with Contextual Attention

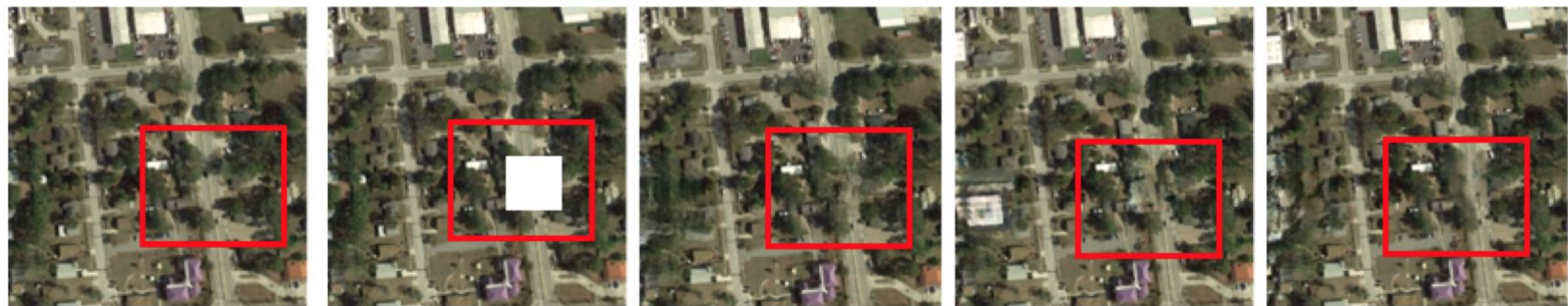
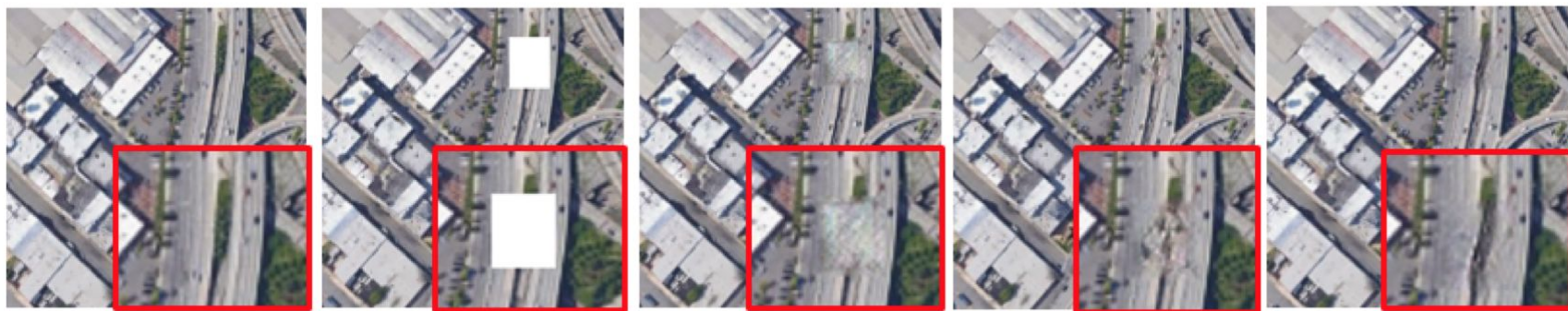
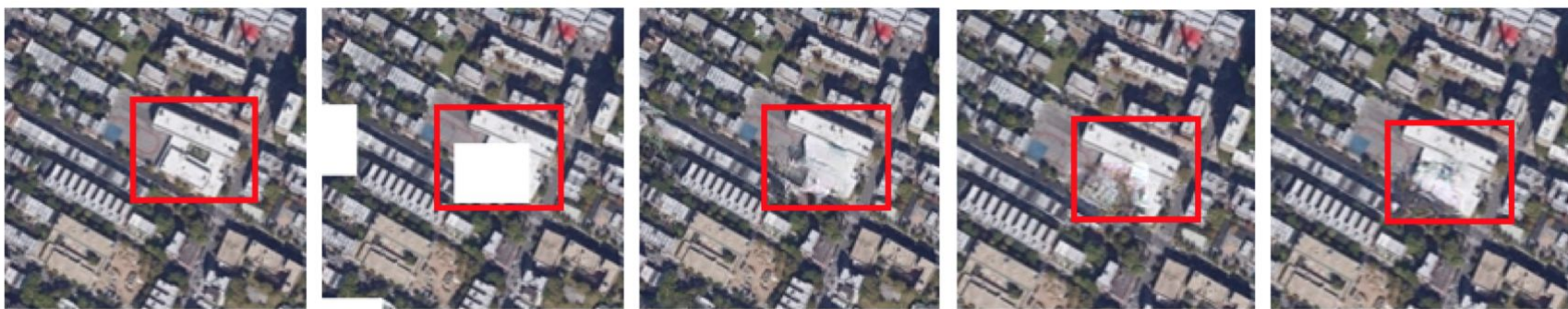
[CVPR 2018 Paper](#) | [ArXiv](#) | [Project](#) | [Demo](#) | [YouTube](#) | [BibTex](#)

Update (Jun, 2018):

1. The tech report of our new image inpainting system DeepFillv2 is released. [ArXiv](#) | [Project](#)
2. We also released recorded demo video [YouTube](#) based on DeepFillv1 (CVPR 2018), as well as video [YouTube](#) of DeepFillv2. Best viewed with highest resolution 1080p.
3. DeepFillv1 is trained and mainly works on rectangular masks, while DeepFillv2 can complete images on free-form masks with user guidance as an option.



Example inpainting results of our method on images of natural scene (Places2), face (CelebA) and object (ImageNet). Missing regions are shown in white. In each pair, the left is input image and right is the direct output of our trained generative neural networks without any post-processing.



Original

Input

Pre-trained Model

Model 1

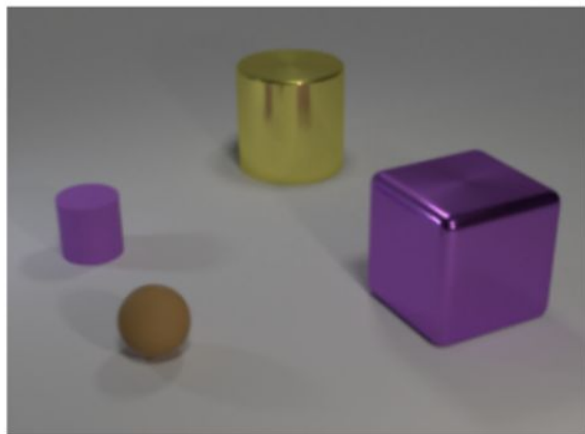
Model 2

기존 알고리즘을 교체

딥러닝으로 풀기 가능해진 문제들

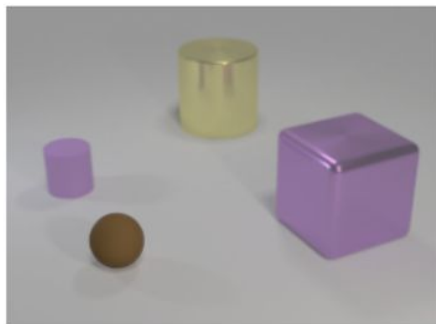
새로운 문제 정의

Original Image:



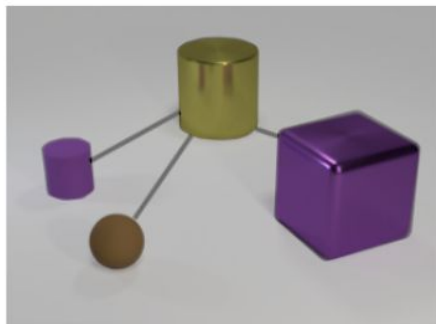
Non-relational question:

What is the size of the brown sphere?



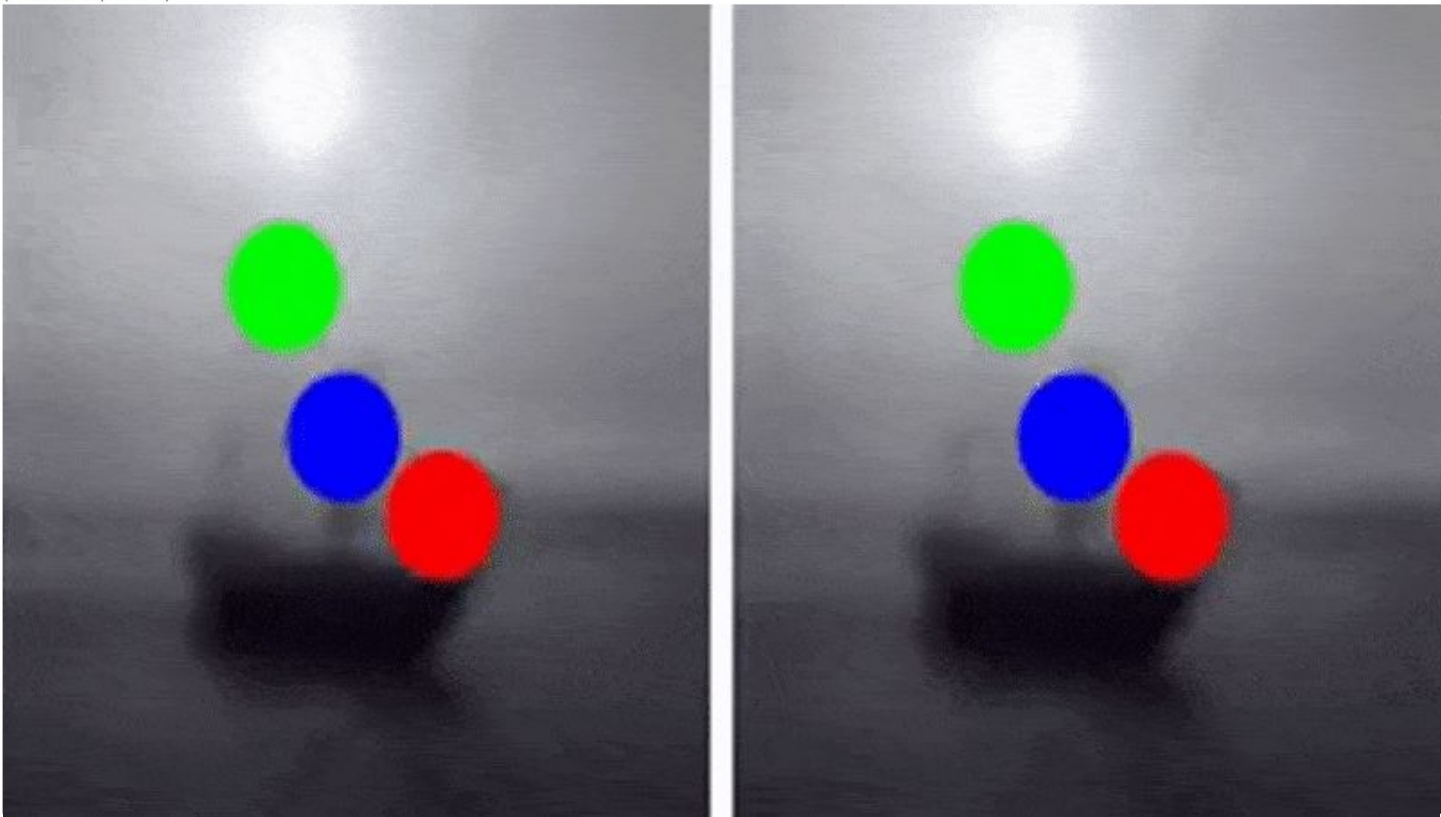
Relational question:

Are there any rubber things that have the same size as the yellow metallic cylinder?



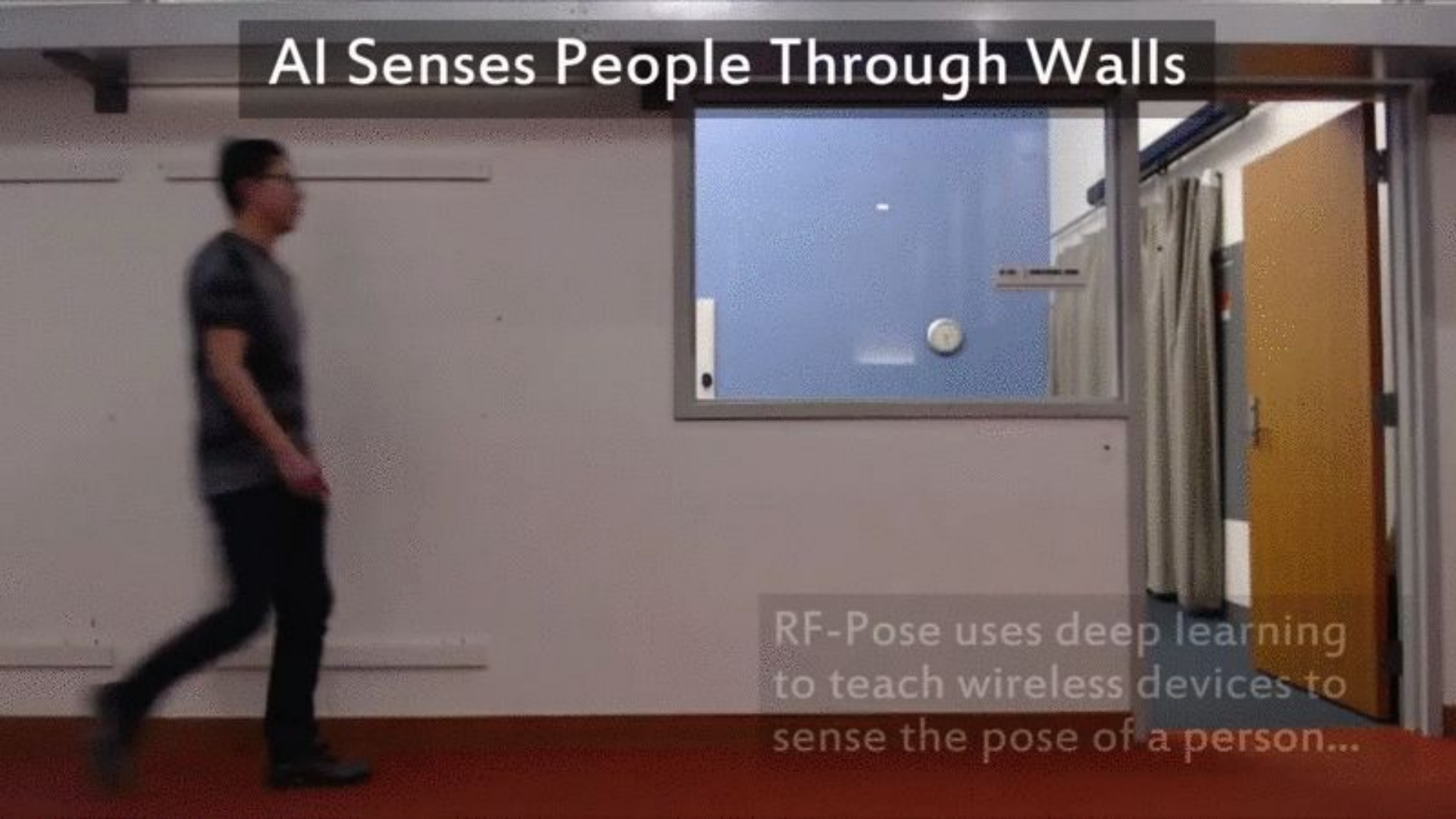
같은 시기에 DeepMind에서 Visual Interaction Networks이란 논문도 나왔습니다. 아래 그림을 보시면 6 프레임을 입력하여 200프레임을 예측한 결과인데, 상당히 잘 맞습니다.

(Ref, DeepMind)



Visual Interaction Network는 시각적 모듈과 물리적 추론 모듈의 두 가지 메커니즘으로 구성되며, 시각적 장면 처리는 물론 미래에 어떤 일이 일어날 지 예측할 수 있는 암묵적 시스템 규칙(예를 들어 물리 시스템)을 학습한다고 합니다. 현상만 잘 학습시키면, 딥러닝 기반의 물리 엔진이 만들어질 기세입니다. 다양한 활용 사례를 기대해봅니다.

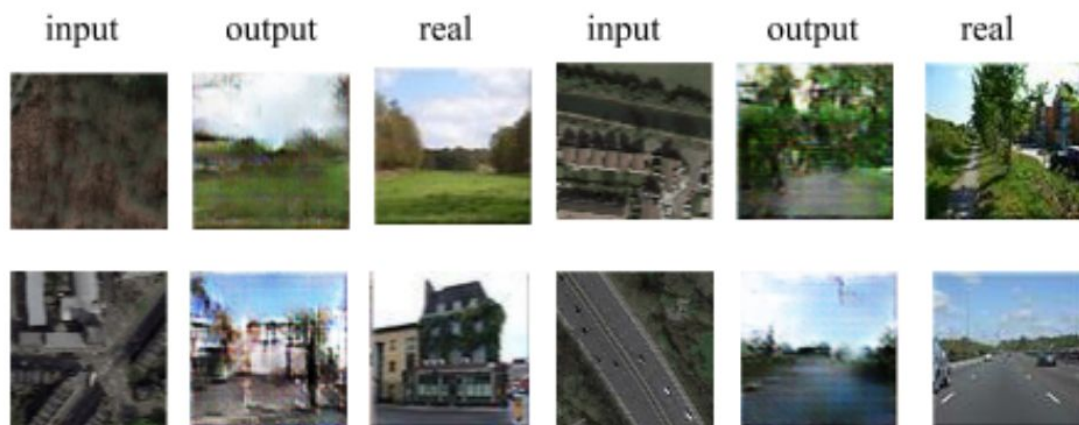
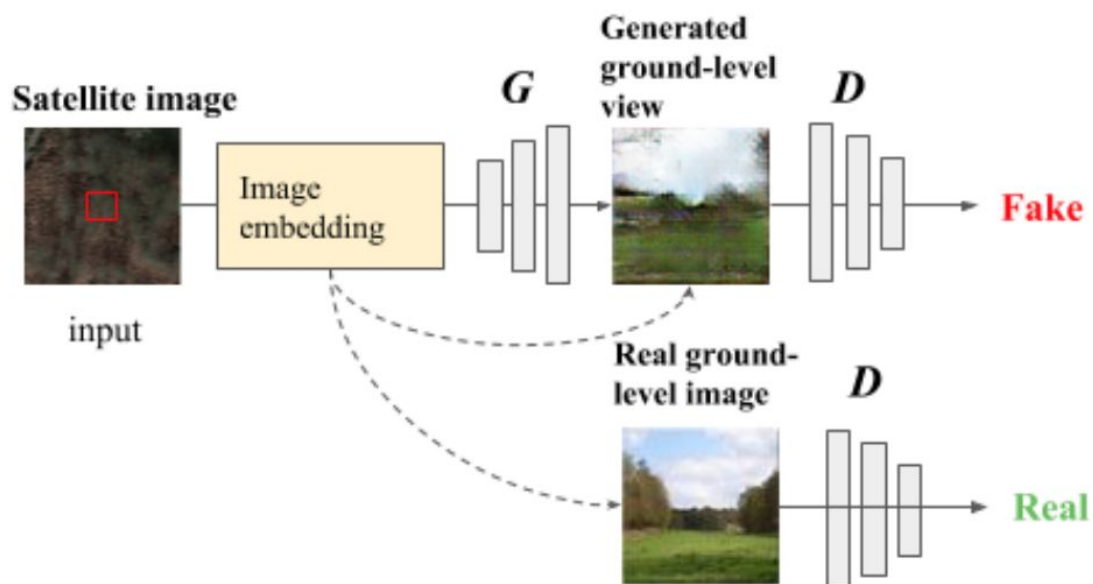
AI Senses People Through Walls

A man in a grey t-shirt and dark pants is walking from left to right in a hallway. The wall is white and has a blue door with a silver handle. To the right, a yellow door is open, revealing a room with a grey curtain. The floor is red.

RF-Pose uses deep learning to teach wireless devices to sense the pose of a **person...**

Given an input video...





Super SloMo: High Quality Estimation of Multiple Intermediate Frames for Video Interpolation

[Huaizu Jiang](#)¹ [Deqing Sun](#)² [Varun Jampani](#)² [Ming-Hsuan Yang](#)^{3,2} [Erik Learned-Miller](#)¹ [Jan Kautz](#)²

¹UMass Amherst ² NVIDIA ³ UC Merced

Abstract

Given two consecutive frames, video interpolation aims at generating intermediate frame(s) to form both spatially and temporally coherent video sequences. While most existing methods focus on single-frame interpolation, we propose an end-to-end convolutional neural network for variable-length multi-frame video interpolation, where the motion interpretation and occlusion reasoning are jointly modeled. We start by computing bi-directional optical flow between the input images using a U-Net architecture. These flows are then linearly combined at each time step to approximate the intermediate bi-directional optical flows. These approximate flows, however, only work well in locally smooth regions and produce artifacts around motion boundaries. To address this shortcoming, we employ another U-Net to refine the approximated flow and also predict soft visibility maps. Finally, the two input images are warped and linearly fused to form each intermediate frame. By applying the visibility maps to the warped images before fusion, we exclude the contribution of occluded pixels to the interpolated intermediate frame to avoid artifacts. Since none of our learned network parameters are time-dependent, our approach is able to produce as many intermediate frames as needed. We use 1,132 video clips with 240-fps, containing 300K individual video frames, to train our network. Experimental results on several datasets, predicting different numbers of interpolated frames, demonstrate that our approach performs consistently better than existing methods.

ORIGINAL SLOMO FOOTAGE



SUPER SLOMO
4X SLOWER



딥러닝의 효용성

사람이 판독한 경우

시뮬레이션의 결과를 정답으로 이용하는 경우

정답이 없는 경우

현실과 유사한 시뮬레이터가 있는 경우

새로운 특징 발견

사람이 판독한 경우

시뮬레이션의 결과를 이용하는 경우

정답이 없는 경우

유의미한 시뮬레이터가 있는 경우

새로운 특징 발견

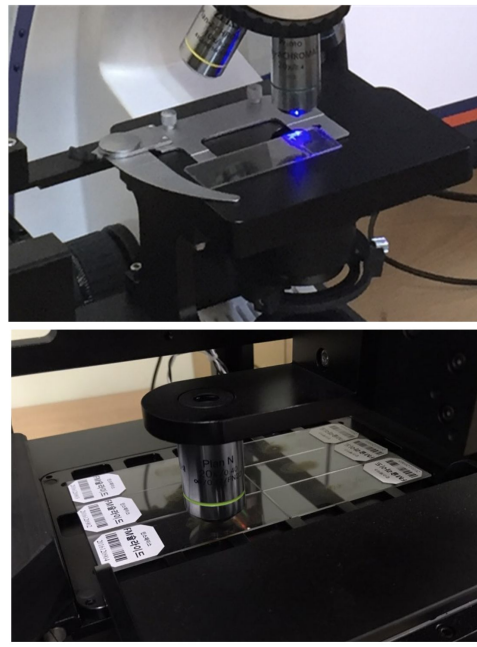
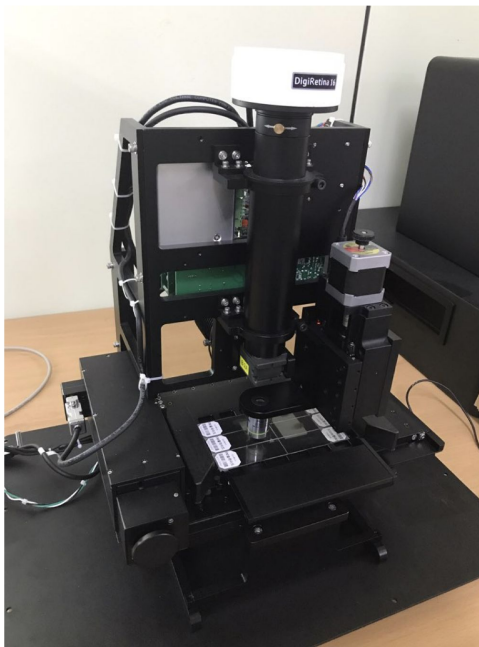
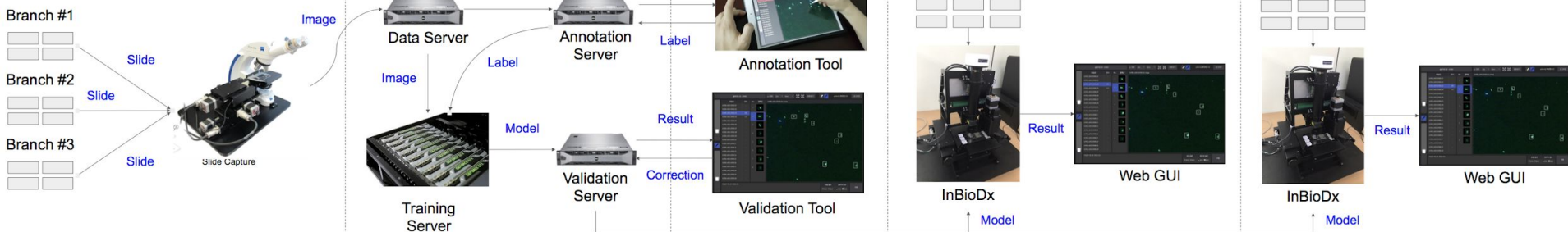


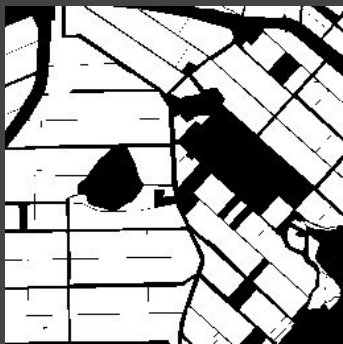
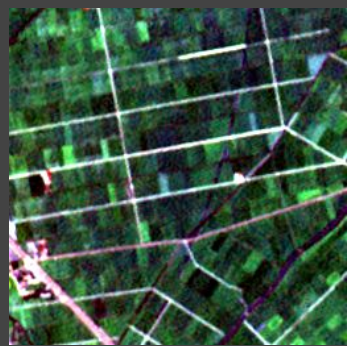
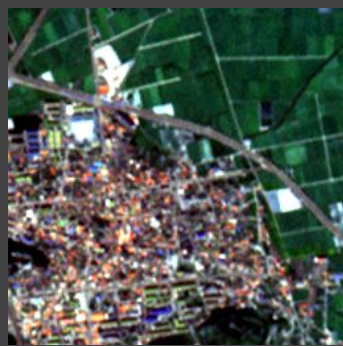
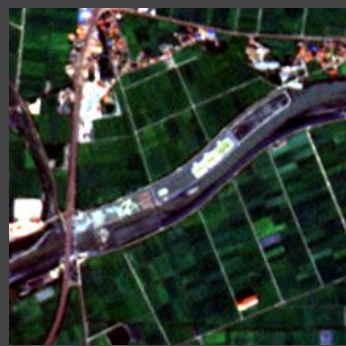
InSpace Insight in Space



Laboratory #1

Laboratory #2





사람이 판독한 경우

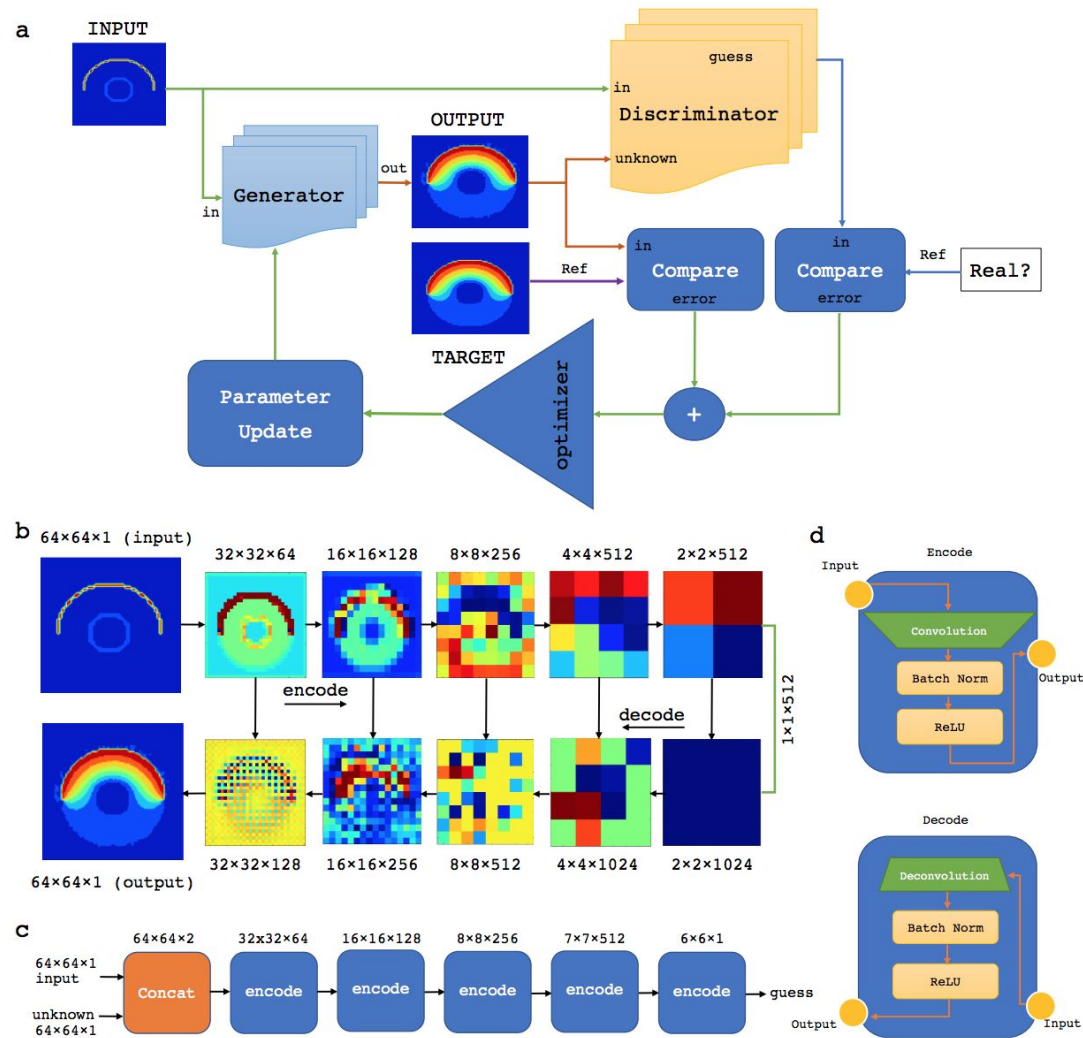
시뮬레이션의 결과를 이용하는 경우

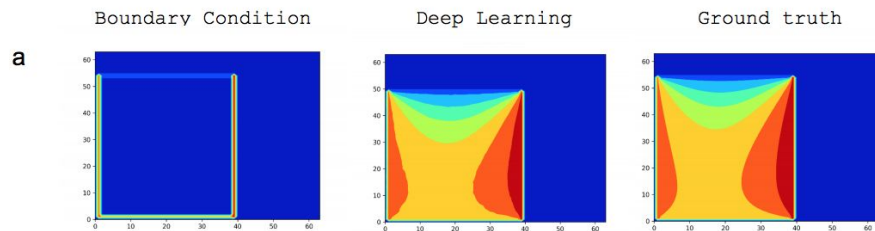
정답이 없는 경우

유의미한 시뮬레이터가 있는 경우

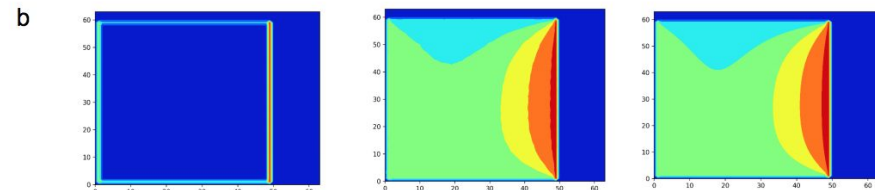
새로운 특징 발견

Deep Learning the Physics of Transport Phenomena

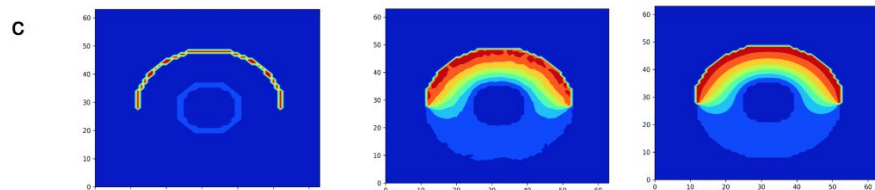




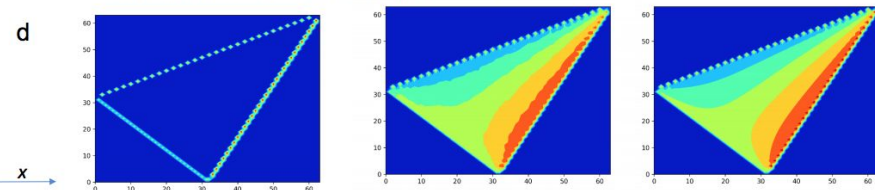
$$T(x, 0) = 60^{\circ}\text{C}; T(x, 54) = 10^{\circ}\text{C}; T(0, y) = 70^{\circ}\text{C}; T(39, y) = 100^{\circ}\text{C}$$



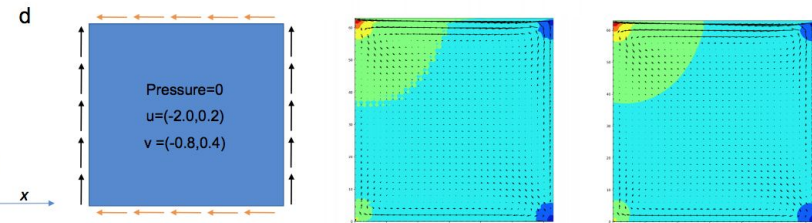
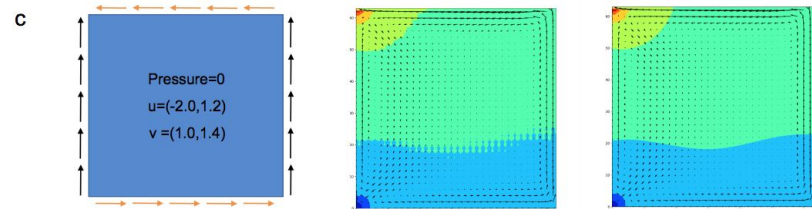
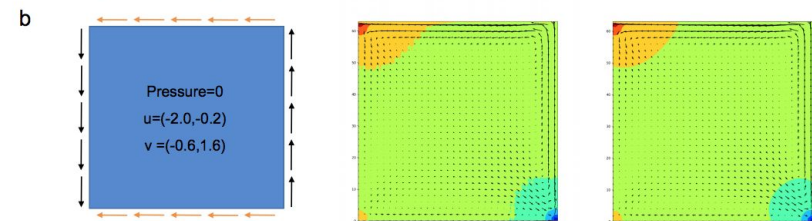
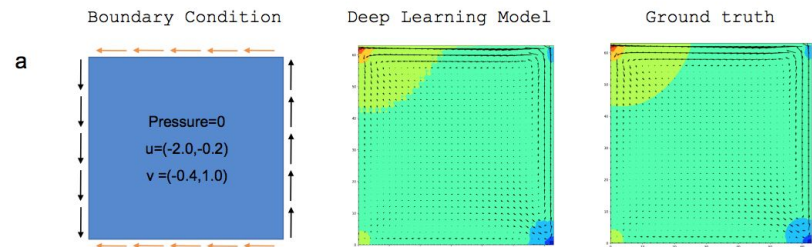
$$T(x, 0) = 20^{\circ}\text{C}; T(x, 59) = 30^{\circ}\text{C}; T(0, y) = 40^{\circ}\text{C}; T(49, y) = 80^{\circ}\text{C}$$



$$T(BC_{\text{outer}}) = (100^{\circ}\text{C}, 0^{\circ}\text{C}); T(BC_{\text{inner}}) = (0^{\circ}\text{C}, 0^{\circ}\text{C})$$



$$T(BC) = (50 \times \sin(x+y)^{\circ}\text{C}, 75 \times \cos(x+y)^{\circ}\text{C}, 100 \times |\sin(x+y)|^{\circ}\text{C})$$



사람이 판독한 경우

시뮬레이션의 결과를 이용하는 경우

정답이 없는 경우

유의미한 시뮬레이터가 있는 경우

새로운 특징 발견

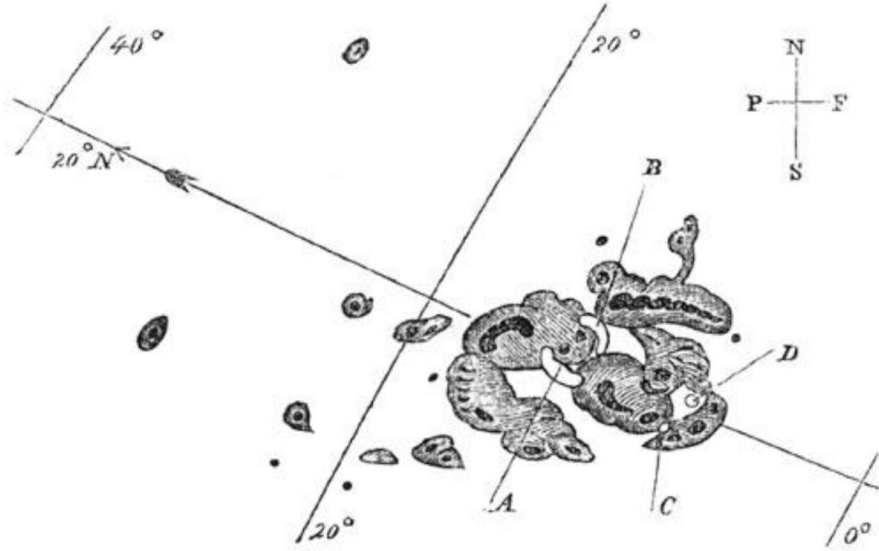
Galileo was drawing sunspots as early as 1613.

400 YEARS **COUNTING
SPOTS**

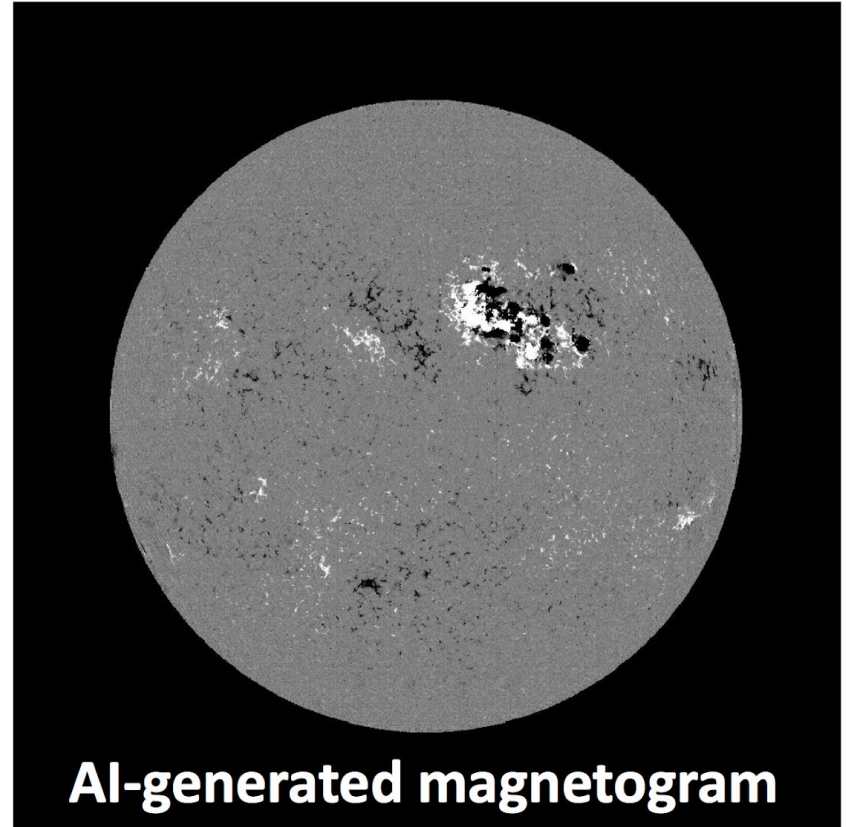
SLOW
SCIENCE



Generation: the Carrington sunspot!



Carrington sunspot in 1859



AI-generated magnetogram

사람이 판독한 경우

시뮬레이션의 결과를 이용하는 경우

정답이 없는 경우

유의미한 시뮬레이터가 있는 경우

새로운 특징 발견

PERFORMANCE
KEEP IT UP!

1505

PTS

TIME 00:05

32 Yds



M. ALONSO

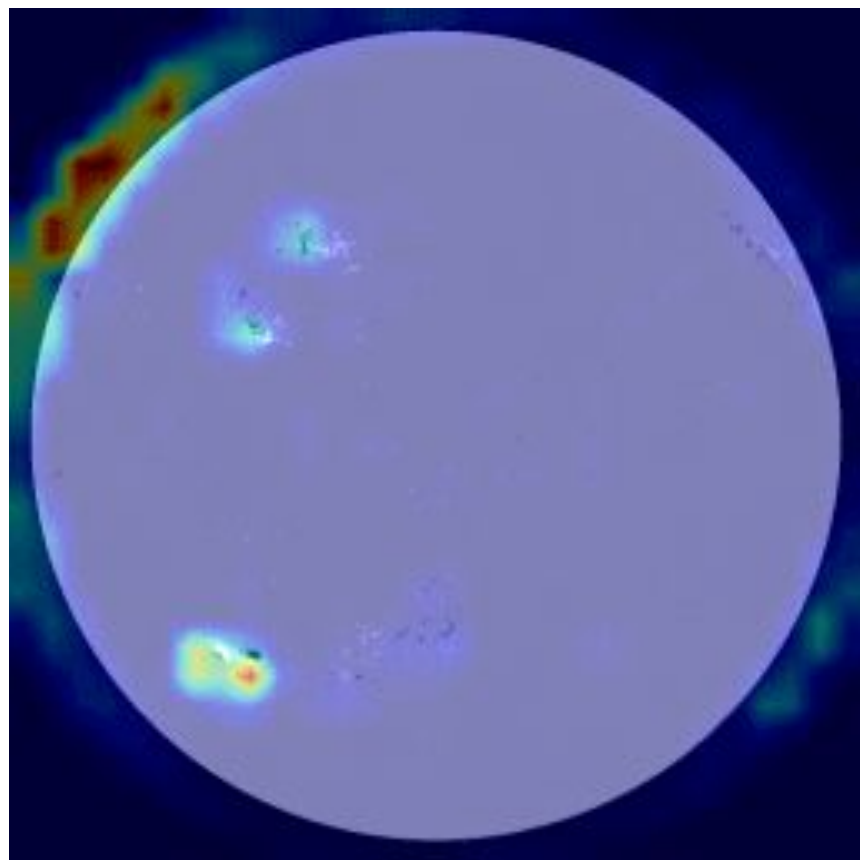
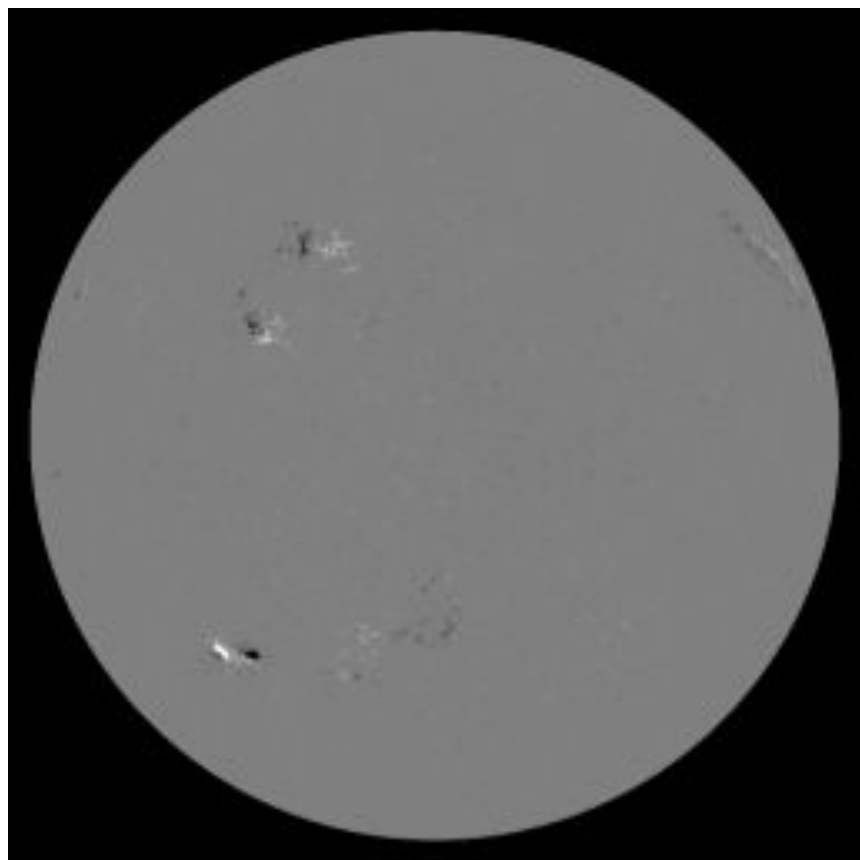
사람이 판독한 경우

시뮬레이션의 결과를 이용하는 경우

정답이 없는 경우

유의미한 시뮬레이터가 있는 경우

새로운 특징 발견



딥러닝 적용 실무

논문 공개 (아카이브, arXiv)

코드 공개 (깃허브, GitHub)

튜토리얼, 블로그, 책, 세미나

노트북에서 서버까지, 일반 게임용 **GPU**도 가능

```
Epoch 37/50
18750/18750 [=====] - 0s - loss: 0.5334 - acc: 0.7354
Epoch 38/50
18750/18750 [=====] - 0s - loss: 0.5305 - acc: 0.7354
Epoch 39/50
18750/18750 [=====] - 0s - loss: 0.5267 - acc: 0.7378
Epoch 40/50
18750/18750 [=====] - 0s - loss: 0.5192 - acc: 0.7466
Epoch 41/50
18750/18750 [=====] - 0s - loss: 0.5237 - acc: 0.7422
Epoch 42/50
18750/18750 [=====] - 0s - loss: 0.5159 - acc: 0.7458
Epoch 43/50
18750/18750 [=====] - 0s - loss: 0.5145 - acc: 0.7494
Epoch 44/50
18750/18750 [=====] - 0s - loss: 0.5073 - acc: 0.7557
Epoch 45/50
18750/18750 [=====] - 0s - loss: 0.5038 - acc: 0.7579
Epoch 46/50
18750/18750 [=====] - 0s - loss: 0.4987 - acc: 0.7629
Epoch 47/50
18750/18750 [=====] - 0s - loss: 0.4980 - acc: 0.7601
Epoch 48/50
18750/18750 [=====] - 0s - loss: 0.4922 - acc: 0.7664
Epoch 49/50
18750/18750 [=====] - 0s - loss: 0.4910 - acc: 0.7637
Epoch 50/50
18750/18750 [=====] - 0s - loss: 0.4877 - acc: 0.7686
[INFO] evaluating on testing set...
6250/6250 [=====] - 0s
[INFO] loss=0.6029, accuracy: 67.3760%
```

데이터셋 구축 - Annotation Tool

모델 검증 - Validation Tool

모델 성능 모니터링

재모델링 자동화

가치 창출 - 이로운 가치를 줄 수 있는가?

수익성 - 경제적인 기대효과가 있는가?

지속성 - 지속적으로 유지될 수 있는가?

6,148,72,35,0,33.6,0.627,50,1
 1,85,66,29,0,26.6,0.351,31,0
 8,183,64,0,0,23.3,0.672,32,1
 1,89,66,23,94,28.1,0.167,21,0
 0,137,40,35,168,43.1,2.288,33,1
 5,116,74,0,0,25.6,0.201,30,0
 3,78,50,32,88,31.0,0.248,26,1
 10,115,0,0,0,35.3,0.134,29,0
 2,197,70,45,543,30.5,0.158,53,1
 8,125,96,0,0,0.0,0.232,54,1
 4,110,92,0,0,37.6,0.191,30,0
 10,168,74,0,0,38.0,0.537,34,1
 10,139,80,0,0,27.1,1.441,57,0
 1,189,60,23,846,30.1,0.398,59,1
 5,166,72,19,175,25.8,0.587,51,1
 7,100,0,0,0,30.0,0.484,32,1
 0,118,84,47,230,45.8,0.551,31,1
 7,107,74,0,0,29.6,0.254,31,1

UCI



Machine Learning Repository

Center for Machine Learning and Intelligent Systems

Pima Indians Diabetes Data Set

Download: [Data Folder](#), [Data Set Description](#)

- 인스턴스 수 : 768개
- 속성 수 : 8가지
- 클래스 수 : 2가지

8가지 속성(1번~8번)과 결과(9번)의 상세 내용은 다음과 같습니다.

1. 임신 횟수
2. 경구 포도당 내성 검사에서 2시간 동안의 혈장 포도당 농도
3. 이완기 혈압 (mm Hg)
4. 삼두근 피부 두껍 두께 (mm)
5. 2 시간 혈청 인슐린 (μ U/ml)
6. 체질량 지수
7. 당뇨 직계 가족력
8. 나이 (세)
9. 5년 이내 당뇨병이 발병 여부



Pima Indians Diabetes Data Set

Download: [Data Folder](#), [Data Set Description](#)

- 인스턴스 수 : 768개
- 속성 수 : 8가지
- 클래스 수 : 2가지

8가지 속성(1번~8번)과 결과(9번)의 상세 내용은 다음과 같습니다.

1. 임신 횟수
2. 경구 포도당 내성 검사에서 2시간 동안의 혈장 포도당 농도
3. 이완기 혈압 (mm Hg)
4. 삼두근 피부 두껍 두께 (mm)
5. 2 시간 혈청 인슐린 (μ U/ml)
6. 체질량 지수
7. 당뇨 직계 가족력
8. 나이 (세)
9. 5년 이내 당뇨병이 발병 여부

No.	속성	테스트 시간
1	임신 횟수	즉시
2	포도당 내성	지연
3	이완기 혈압	즉시
4	삼두근 피부 두껍 두께	즉시
5	혈청 인슐린	지연
6	체질량 지수	즉시
7	당뇨 직계 가족력	즉시
8	나이	즉시



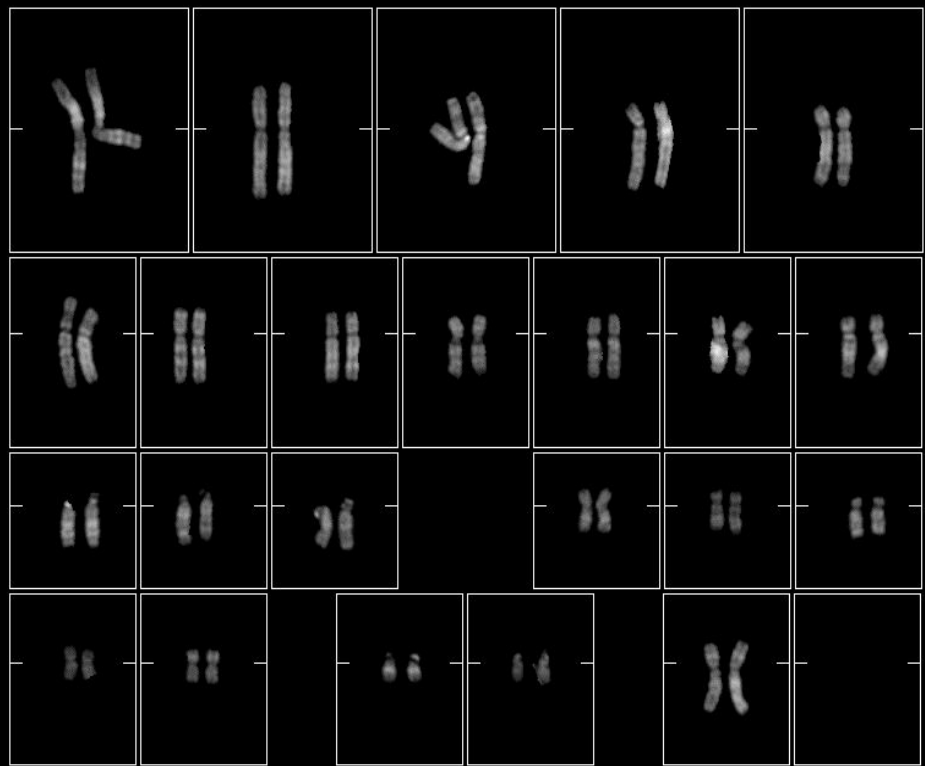
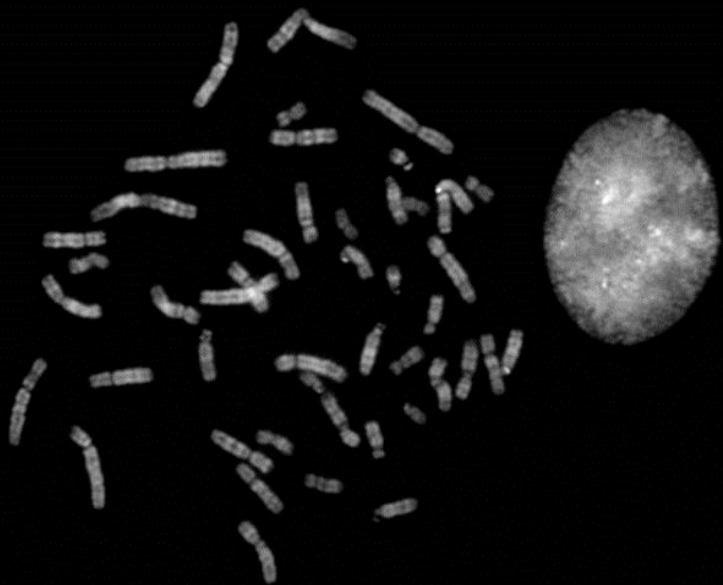
No.	속성	테스트 시간	비용(CAD)	단체할인비용(CAD)
1	임신 횟수	즉시	1.00	N/A
2	포도당 내성	지연	17.61	15.51
3	이완기 혈압	즉시	1.00	N/A
4	삼두근 피부 두겹 두께	즉시	1.00	N/A
5	혈청 인슐린	지연	22.78	20.68
6	체질량 지수	즉시	1.00	N/A
7	당뇨 직계 가족력	즉시	1.00	N/A
8	나이	즉시	1.00	N/A

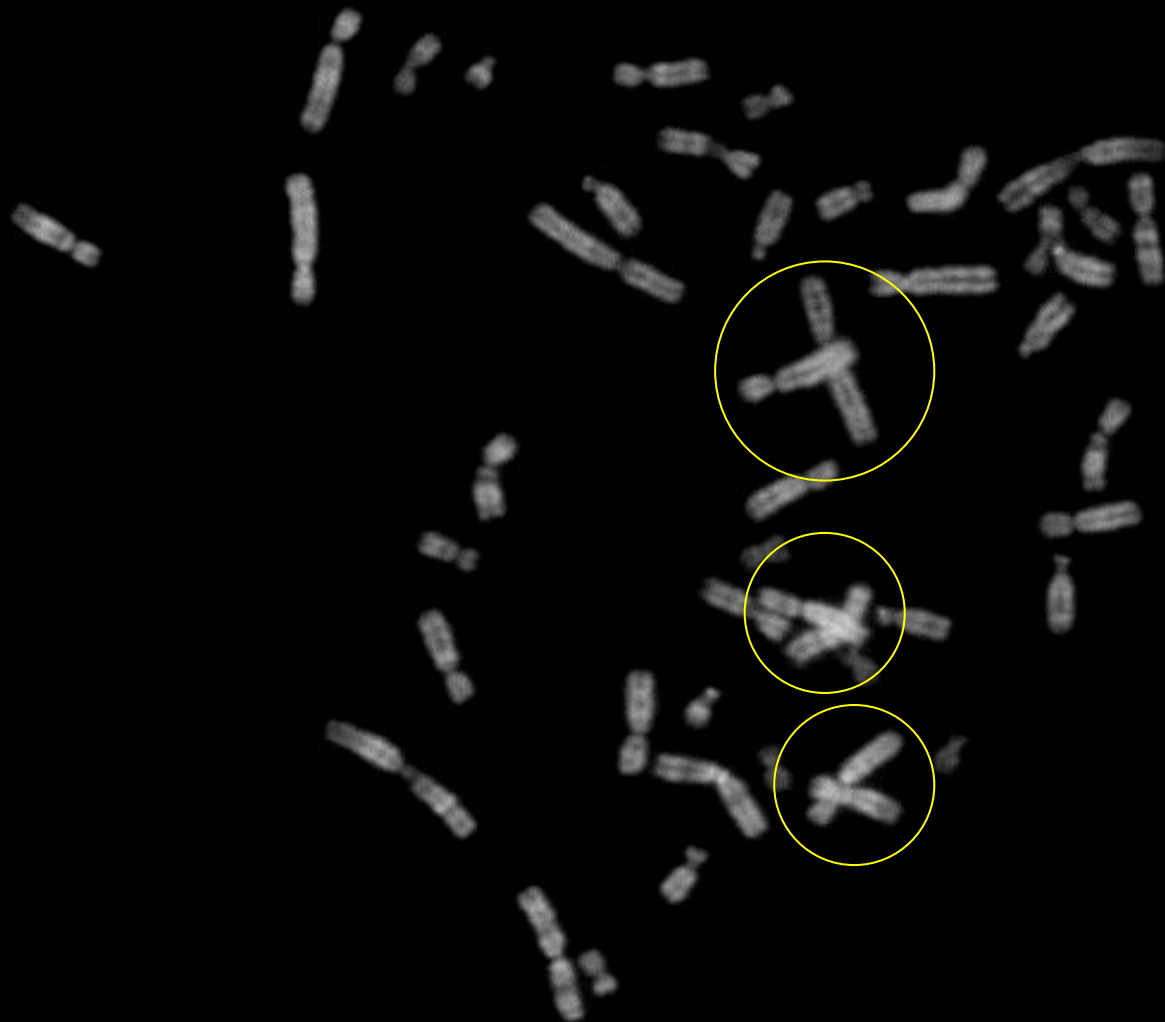
7. 당뇨 직계 가족력

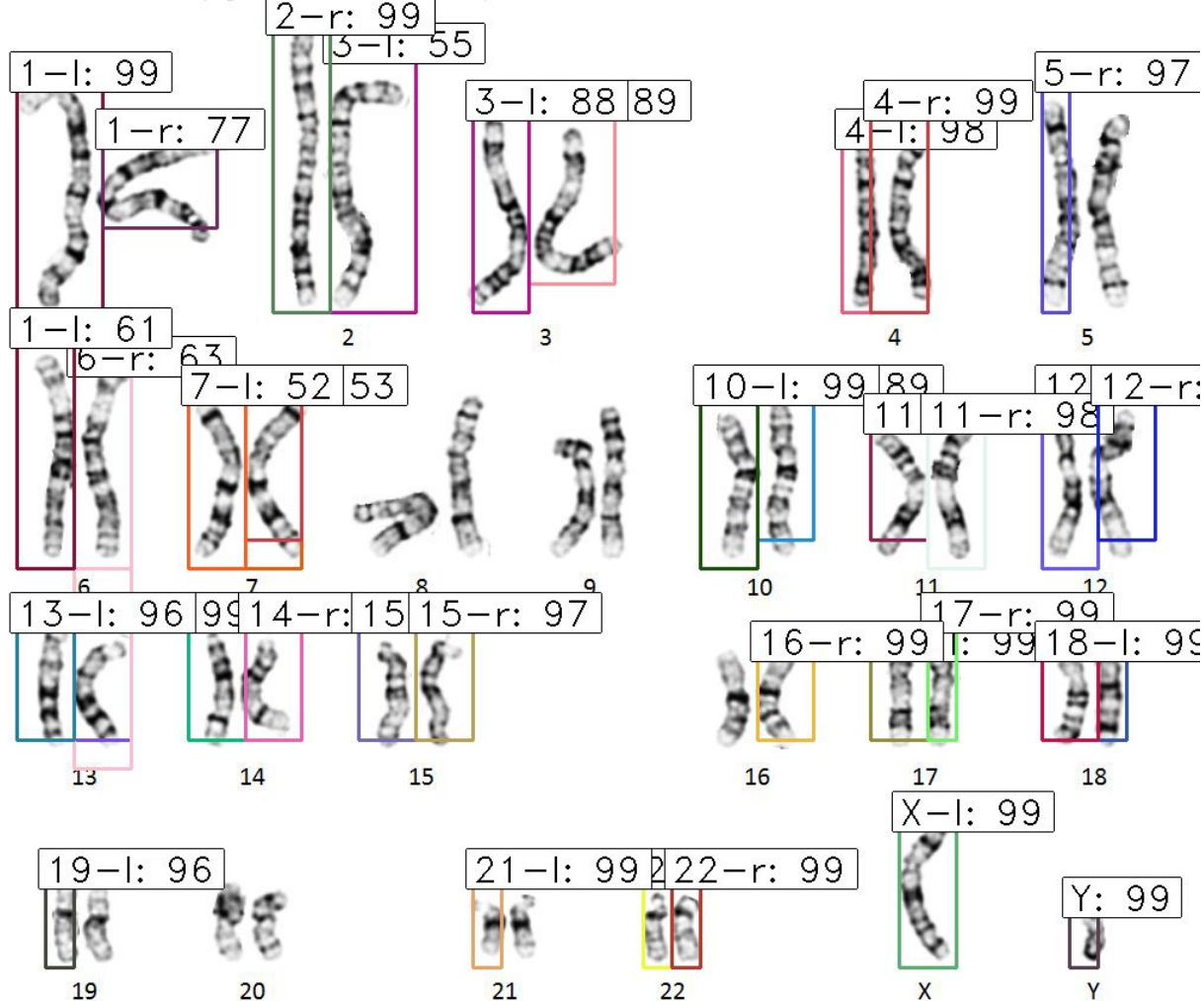
8. 나이 (세)

9. 5년 이내 당뇨병이 발병 여부

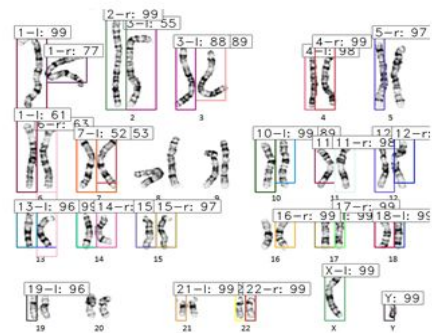
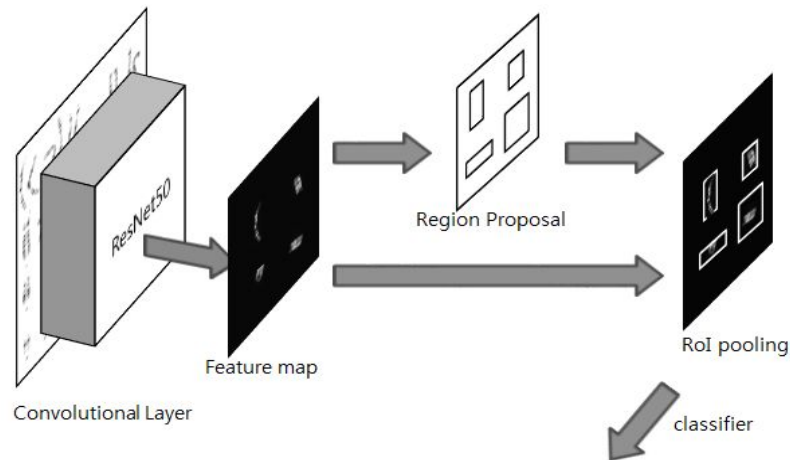








Faster R-CNN

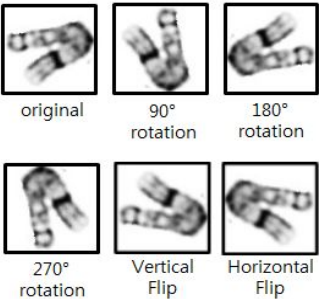


Faster R-CNN Model Architecture

Training Dataset

Label	개수	Label	개수	Label	개수
1-l	40	9-l	40	17-r	40
1-r	40	9-r	40	18-l	40
2-l	40	10-l	40	18-r	40
2-r	40	10-r	40	19-l	40
3-l	40	11-l	40	19-r	40
3-r	40	11-r	40	20-l	40
4-l	40	12-l	40	20-r	40
4-r	40	12-r	40	21-l	40
5-l	39	13-l	40	21-r	40
5-r	39	13-r	40	22-l	40
5	1	14-l	40	22-r	40
6-l	40	14-r	40	X-l	39
6-r	40	15-l	40	X-r	19
7-l	40	15-r	40	X	1
7-r	40	16-l	40	Y	20
8-l	40	16-r	40		
8-r	40	17-l	40	합계	1838

Data augmentation



Hyper Parameter

optimizer	Adamax
learning rate	2e-3
beta_1	0.9
beta_2	0.999
loss function	smooth L1







Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization

This paper comes from Nvidia and goes full throttle on using synthetic data to train Convolutional Neural Networks (CNNs). They created a plugin for Unreal Engine 4 which will generate synthetic training data. The real key is that they randomize many of the variables that training data can have including:

- number and types of objects
- number, types, colors, and scales of distractors
- texture on the object of interest, and background photograph
- location of the virtual camera with respect to the scene
- angle of the camera with respect to the scene
- number and locations of point lights

They showed some pretty promising results that demonstrate the effectiveness of pre-training with synthetic data; a result that previously has not been achieved. It may shed some light on how to go about generating and using synthetic data if you're short on that important resource.

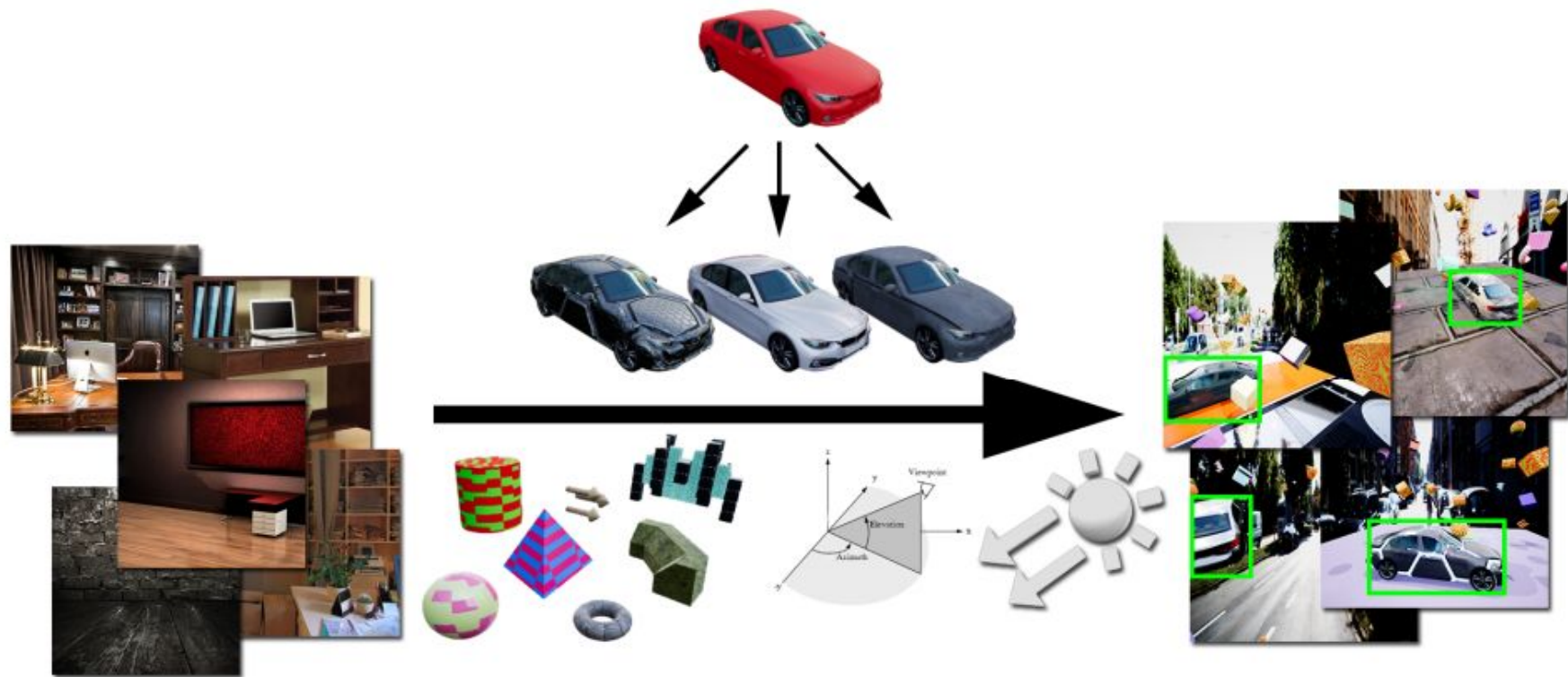
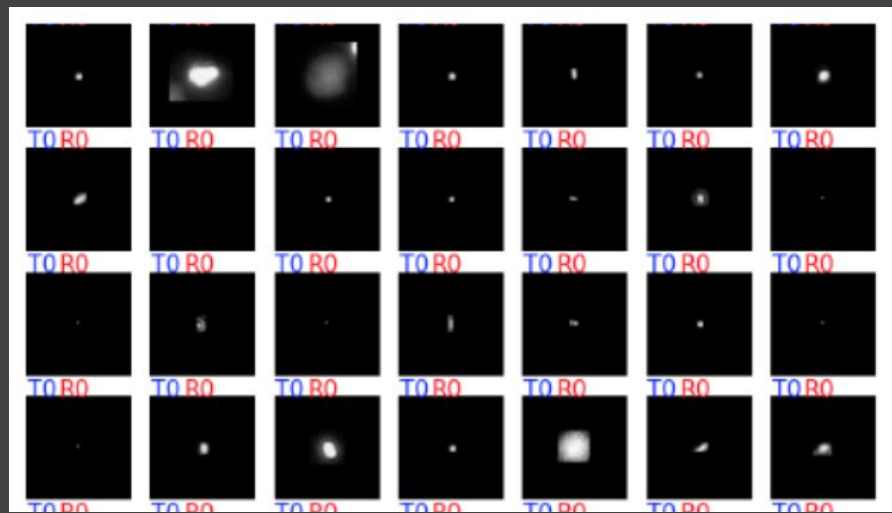
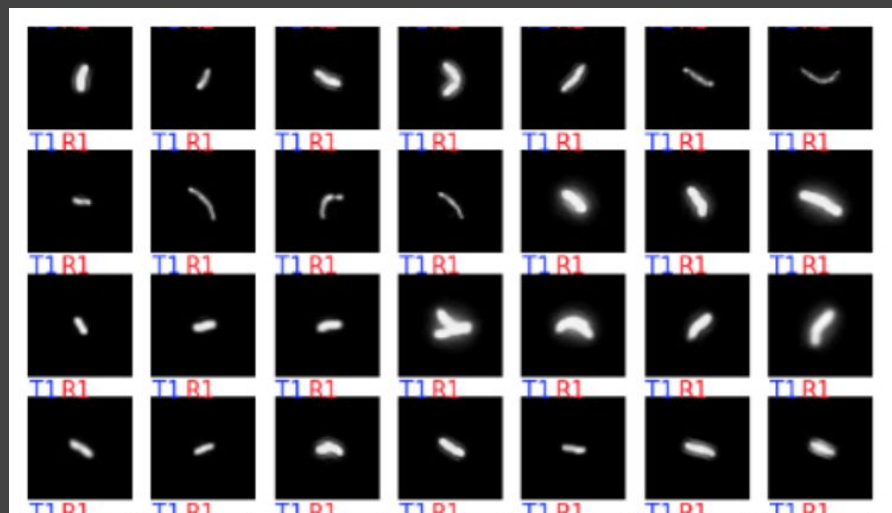
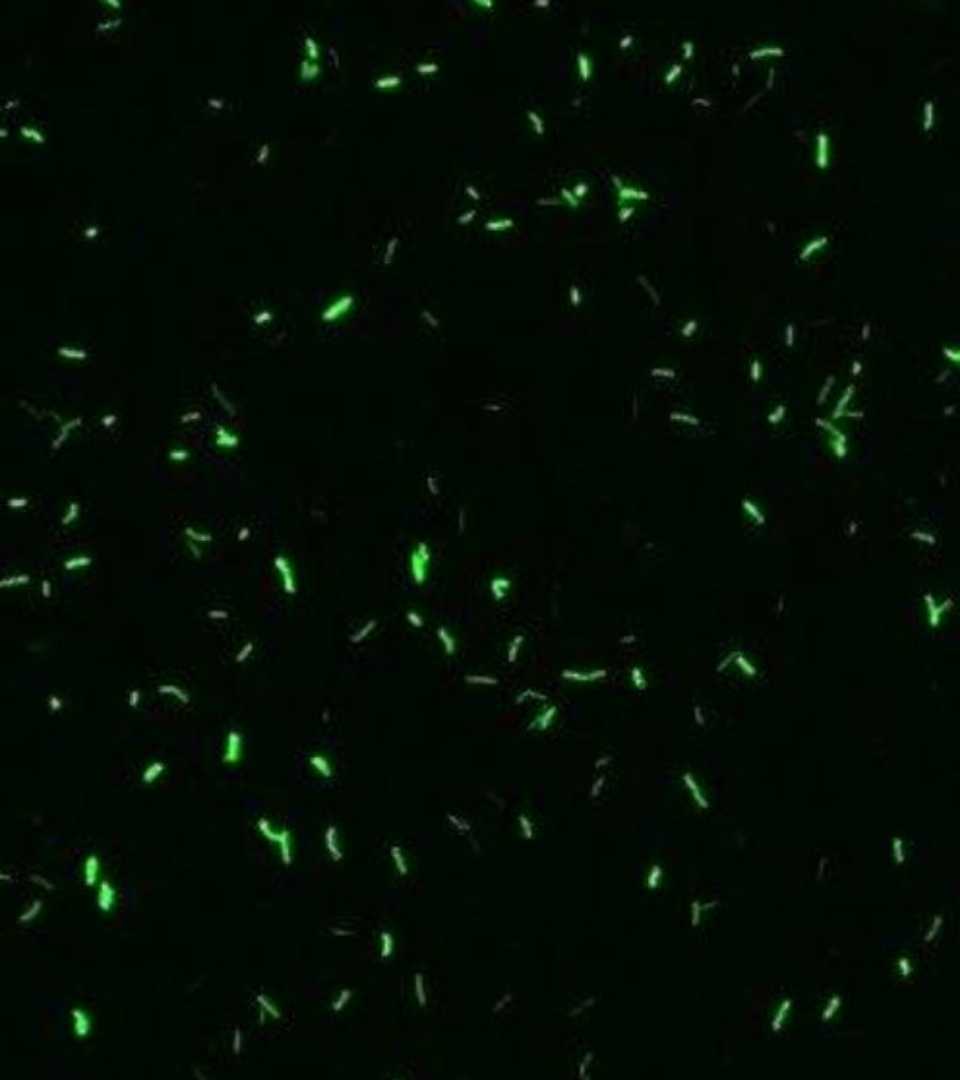
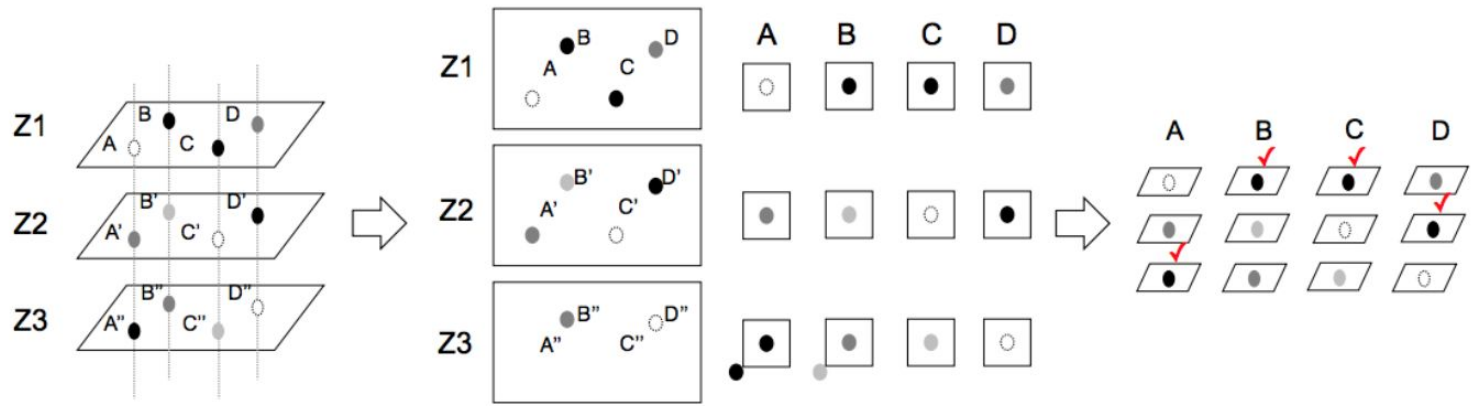
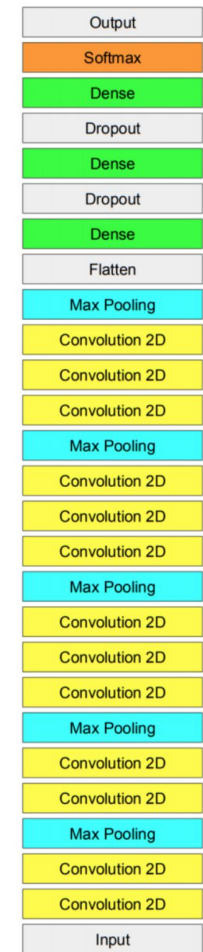


Figure 1. Domain randomization for object detection. Synthetic objects (in this case cars, top-center) are rendered on top of a random background (left) along with random flying distractors (geometric shapes next to the background images) in a scene with random lighting from random viewpoints. Before rendering, random texture is applied to the objects of interest as well as to the flying distractors. The resulting images, along with automatically-generated ground truth (right), are used for training a deep neural network.





Classification : VGG-16



Calculating focus for each object

Class	Train	Validation	Test
TB	75,326	2,084	3,639
Non-TB	155,661	39,495	68,966
Total	230,987	41,579	75,326

Model	Sensitivity	Specificity	Accuracy
AlexNet-like	96%	84%	90%
VGG-16	84%	98.4%	97.7%

Test result with patch images

Model	Sensitivity	Specificity	Accuracy
AlexNet-like	100%	76%	88%
VGG-16	100%	86%	93%

Test result with slides





InSight in Space
InSpace



Laboratory #1

Laboratory #2

Branch #1

Branch #2

Branch #3

Slide

Slide

Slide

Slide Capture

Image

Image

Training
Server

Data Server

Image

Label

Model

Image

Label

Model

Result

Correction

Model

Model

Model

Model

Model

Model

Model

Model

Model

Model

Model

Model

Model

Model

Model

Model

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

Image

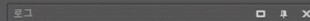
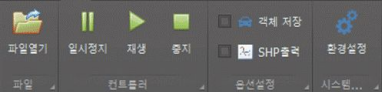
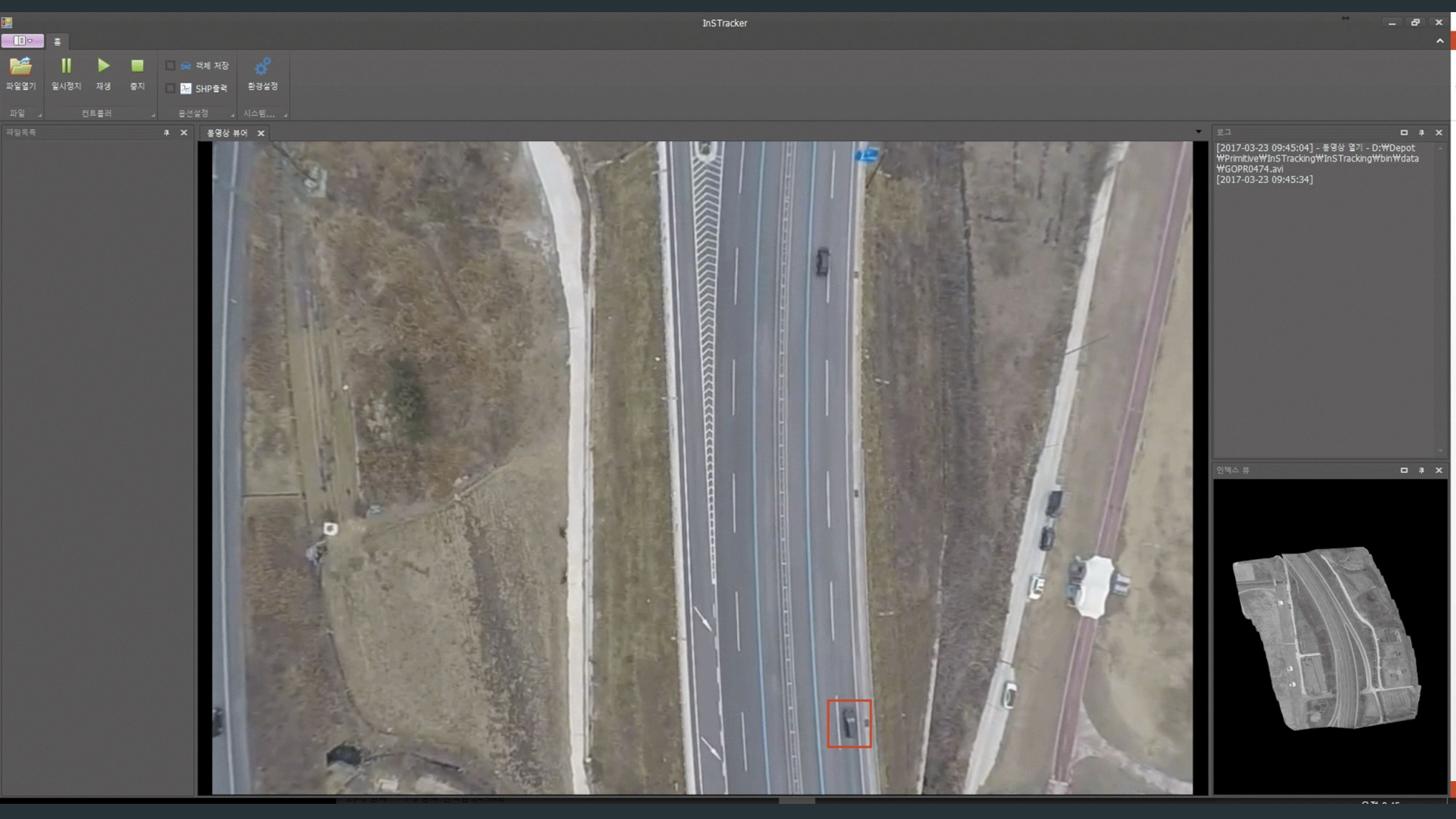
Image

Image

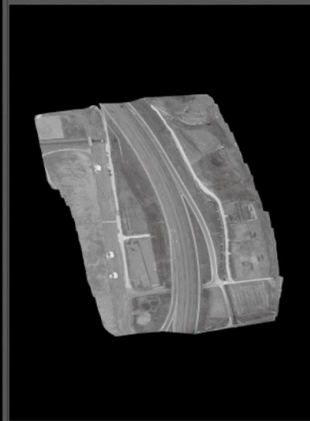
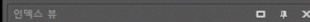








[2017-03-23 09:45:04] - 동영상 열기 - D:\WDepot
WPrimitiveWinTrackingWinTrackingWinWdata
WGOPR0474.avi
[2017-03-23 09:45:34]



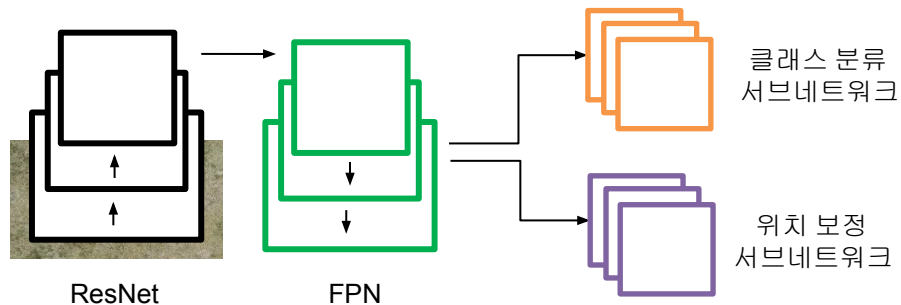








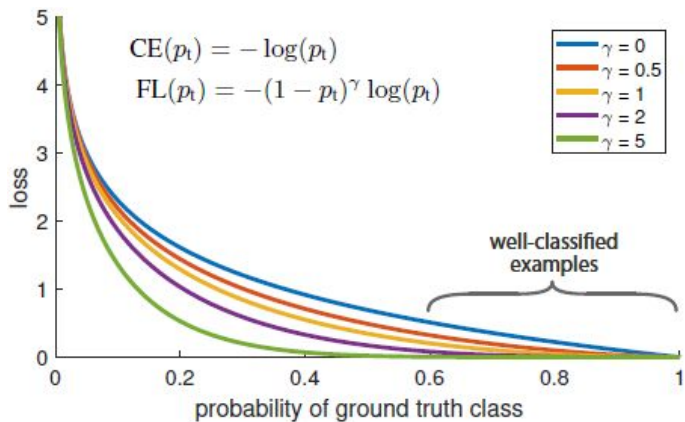
RetinaNet with ResNet50



RetinaNet 모델 구성

버스	413	훈련셋	5601
승용차	2437	검증셋	1102
전차	2678		
트럭	1175	합계	6703

데이터셋 구성



Focal Loss 함수

초기 학습률	1e-5
손실 함수	smooth_l1, focal
최적화 함수	adam

하이퍼 파라미터 구성



산림보호

산불

병해충

▪ 산림병해충

- 산림해충

- 산림병해

- 소나무재선충병

- 병해충 식별법

- 산림청활동내역

▪ 병해충검색

▪ 예찰진단

▪ 방제정보

▪ 발생신고·상담

백두대간

산림유전자원보호구역

나고야의정서(ABS)

숲사람

> 분야별 산림정보 > 산림보호 > 병해충 > 산림병해충 > 소나무재선충병

소나무재선충병



프린트

개요

▶ 소나무재선충병이란?

소나무재선충은 크기 1mm 내외의 실같은 선충으로서매개충(솔수염하늘소, 북방수염하늘소)의 몸 안에 서식하다가 새순을 갇아 먹을때 상처부위를 통하여 나무에 침입한다. 침입한 재선충은 빠르게 증식하여 수분, 양분의 이동통로를 막아 나무를 죽게 하는 병으로 치료약이 없어 감염되면 100% 고사한다.

■ 재선충 침입 6일째부터 잎이 처지고 20일째에 잎이 시들기 시작하여 30일 후 잎이 급속하게 붉은 색으로 변색하며 고사

■ 소나무재선충병에 감염되는 수종 : 소나무, 해송, 잣나무

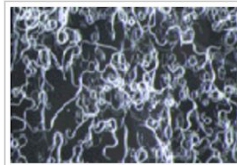


감염되지 않은 건강한 소나무



재선충에 감염된 소나무

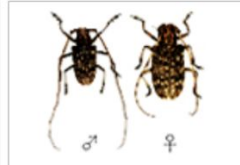
▶ 소나무재선충 사진자료



소나무 재선충 현미경사진



솔수염 하늘소 성충



솔수염 하늘소



촬영장비 : 고정익 무인기
공간해상도 : 2.54cm
채널 : RGB & NIR





[고정익 무인기]

무게 : 2.25kg

배터리 : 8000mAh(LiPo)

공간해상도 : 2.54cm(@400ft)

채널 : RGB & NIR

속도 : 23m/s

[드론]

무게 : 1.160kg

배터리 : 5200mAh(LiPo)

카메라화소 : 1400만

채널 : RGB

속도 : 15m/s

[드론]

무게 : 1.242kg

배터리 : 5200mAh(LiPo)

카메라화소 : 1400만

채널 : RGB

속도 : 15m/s



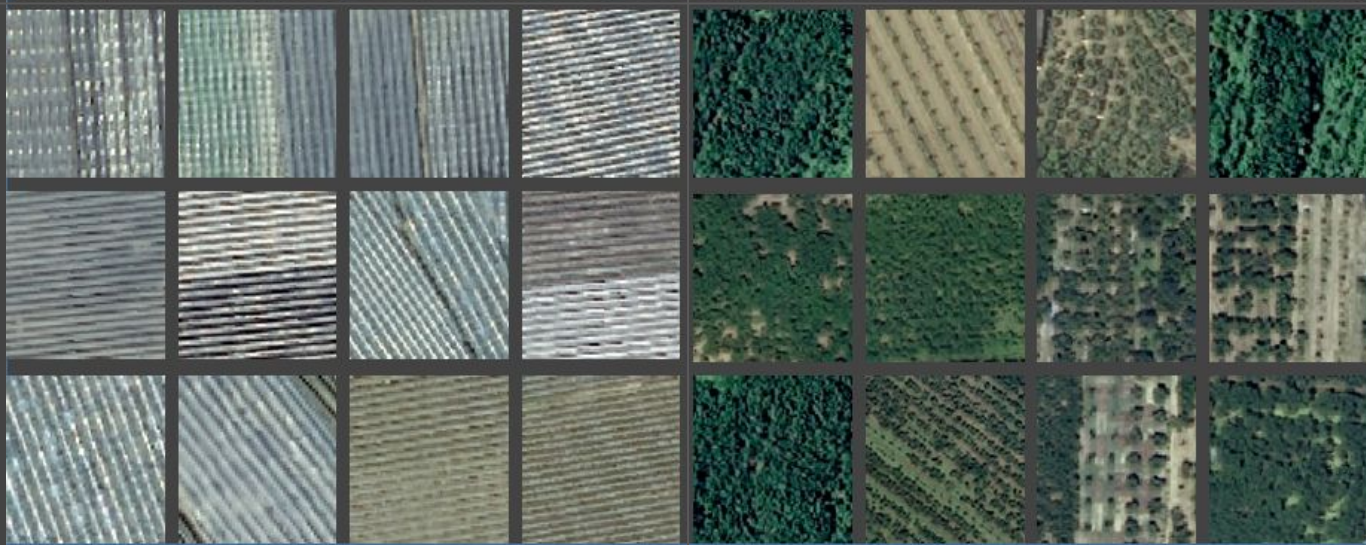


사과

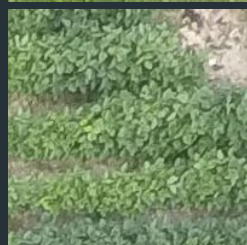
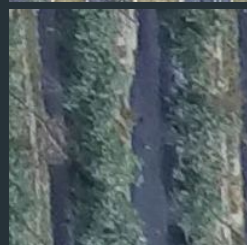
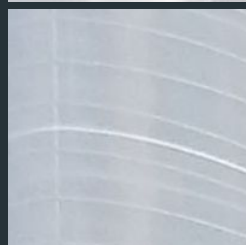
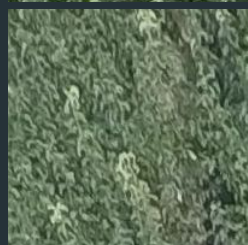
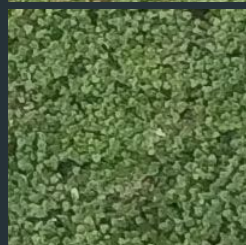
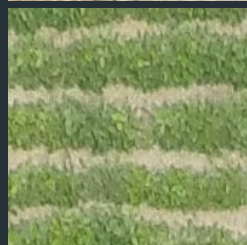
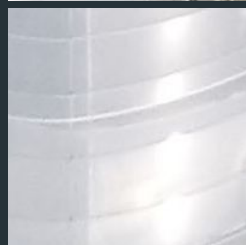
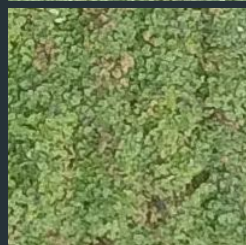
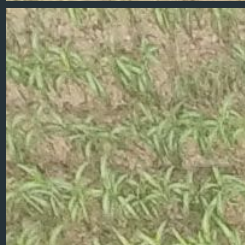
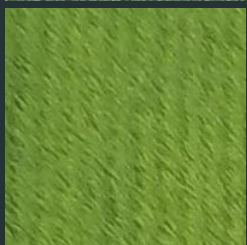
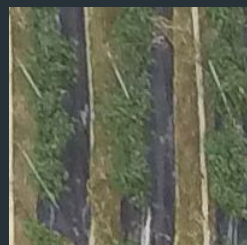
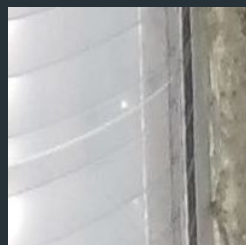
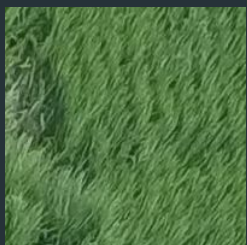
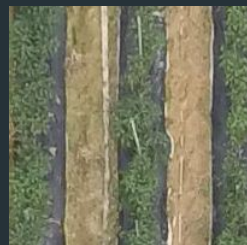
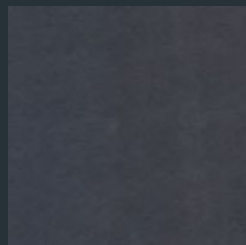
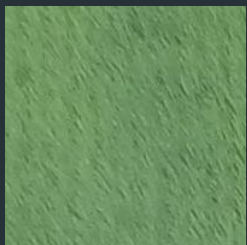


배

포도



복숭아



벼

옥수수

고구마

과수

비닐하우스

고추

깻잎



20M



30M



40M







RapidEye

Delivering the World

Mission Characteristic

Number of Satellites

Spacecraft Lifetime

Orbit Altitude

Equator Crossing Time

Sensor Type

Spectral Bands

Ground Sampling Distance (nadir)

Pixel size (orthorectified)

Swath Width

Information

5

Over 7 years

630 km in Sun-synchronous orbit

11:00 am local time (approximately)

Multi-spectral push broom imager

Capable of capturing all of the following spectral bands:

Band Name	Spectral Range (nm)
Blue	440 – 510
Green	520 – 590
Red	630 – 685
Red Edge	690 – 730
NIR	760 – 850

6.5 m

5 m

77 km



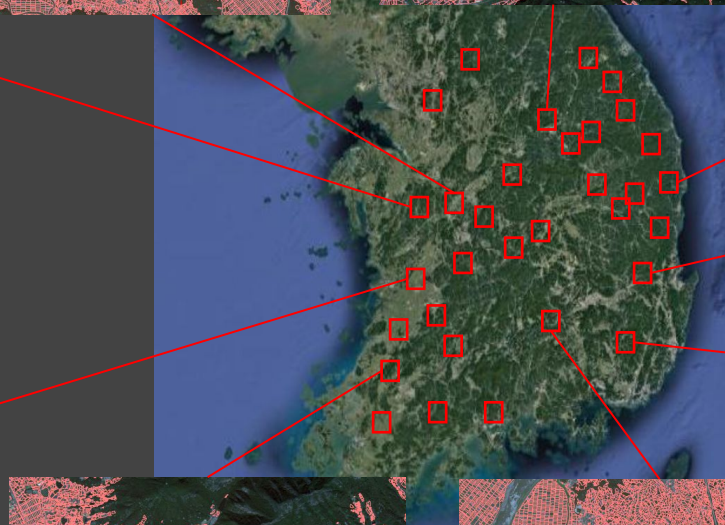
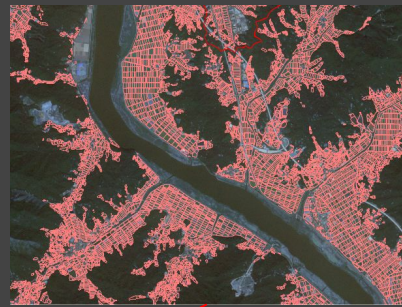
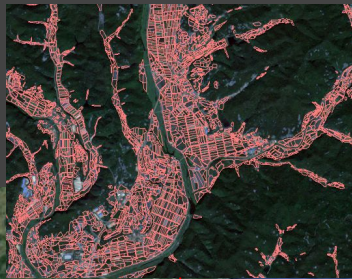
Table of Contents

- Layers
 - ☒ Digitizing_anak
 - ☒ anak_mosaic.img
 - RGB
 - Red: Layer_1
 - Green: Layer_2
 - Blue: Layer_3

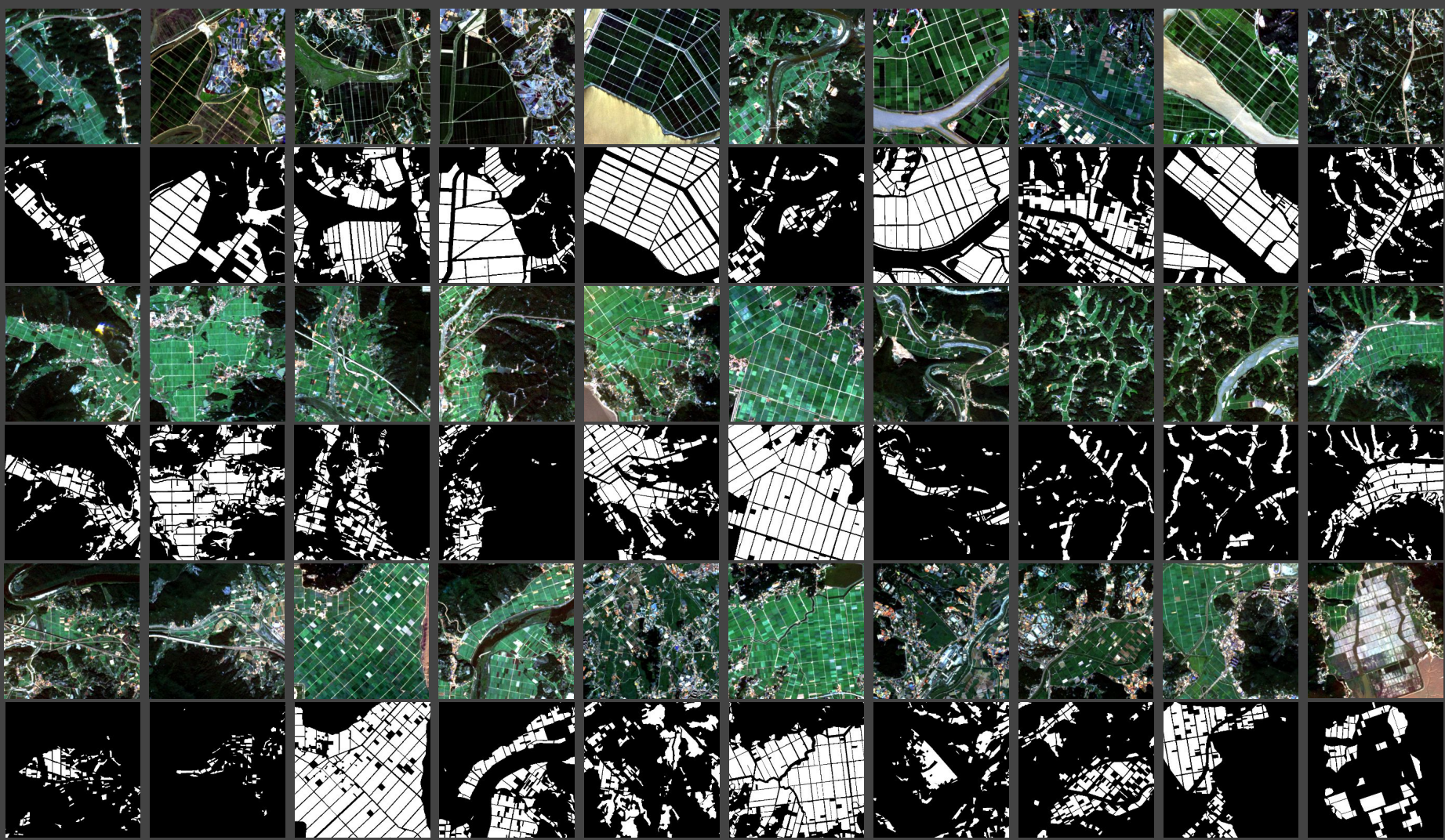
ArcToolbox

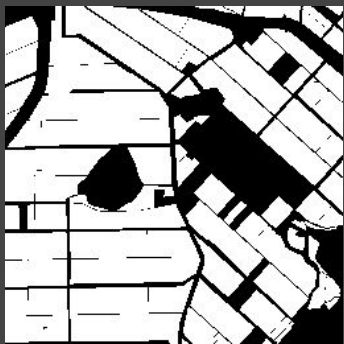
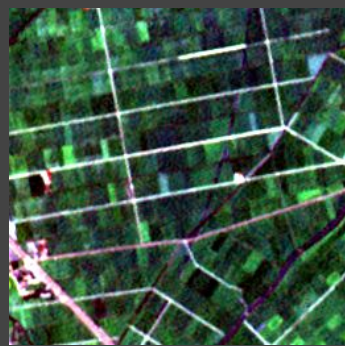
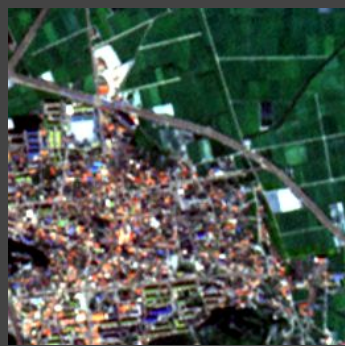
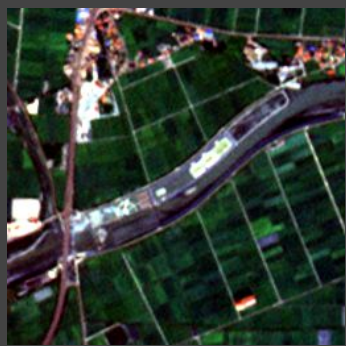
- ArcToolbox
 - 3D Analyst Tools
 - Analysis Tools
 - Cartography Tools
 - Conversion Tools
 - Data Interoperability Tools
 - Data Management Tools
 - Editing Tools
 - Geocoding Tools
 - Geostatistical Analyst Tools
 - Linear Referencing Tools
 - Multidimension Tools
 - Network Analyst Tools
 - Parcel Fabric Tools
 - Schematics Tools
 - Server Tools
 - Spatial Analyst Tools
 - Spatial Statistics Tools
 - Tracking Analyst Tools



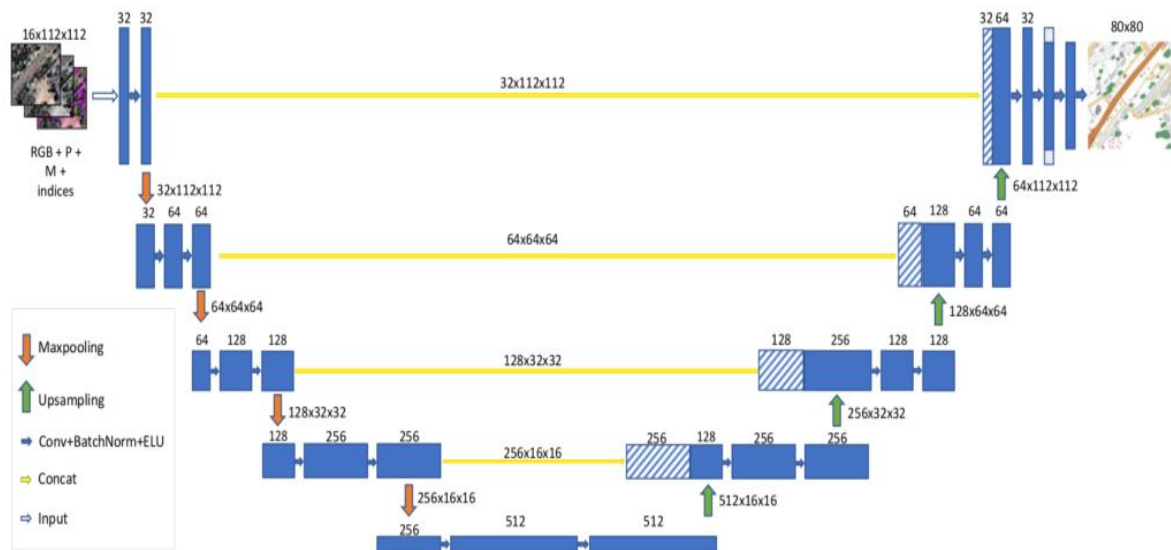








U-Net



Model Architecture

Binary classification accuracy

Pixel Accuracy	0.9480
Mean IOU	0.8035

Data Set info.

Type	Satellite Image
Shape	(256, 256, 5)
Total number of data	5,604
Ratio(Train,Val,Test)	7:2:1
Data capacity	3.78G

Hyperparameter info.

Activation	Relu/sigmoid(last)
Padding	Same
Batch_size	36
Epoch	100
Optimizer	Adam
Loss func.	Iou_loss





PolygonRNN++

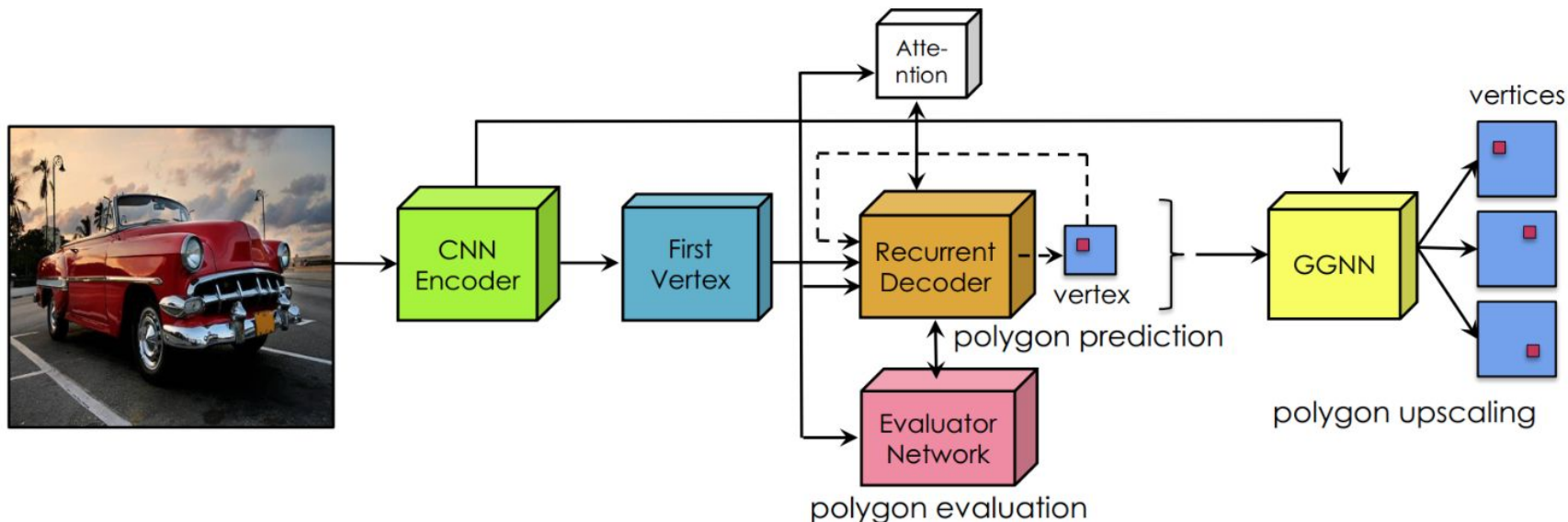
This is the official repository for the Polygon-RNN++ project (CVPR-2018). For technical details, please refer to:

Efficient Interactive Annotation of Segmentation Datasets with Polygon-RNN++

David Acuna*, Huan Ling*, Amlan Kar*, Sanja Fidler (* denotes equal contribution)

CVPR 2018

[\[Paper\]](#) [\[Video\]](#) [\[Project Page\]](#) [\[Demo\]](#)



PolygonRNN++ (with GT boxes)



0 clicks



0 clicks



0 clicks

Human Annotator



576 clicks



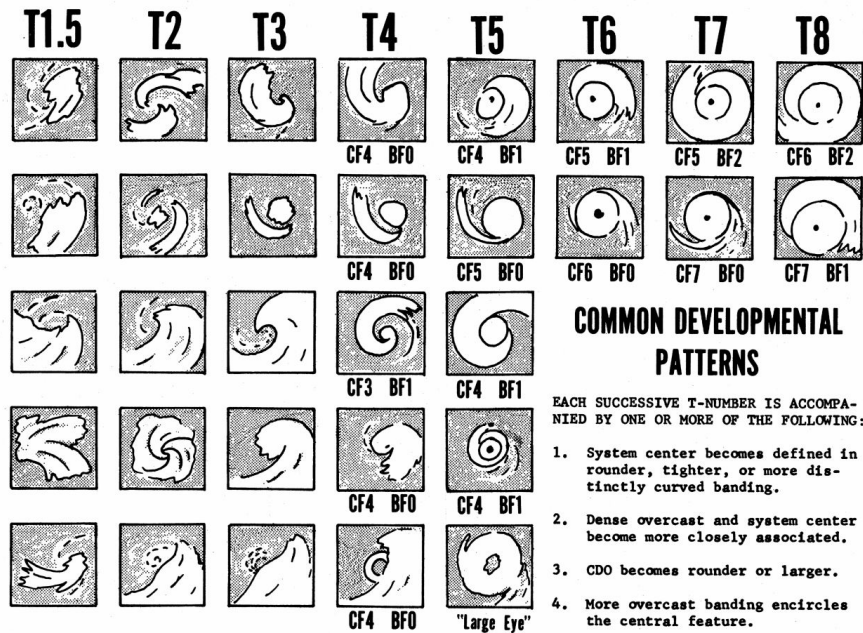
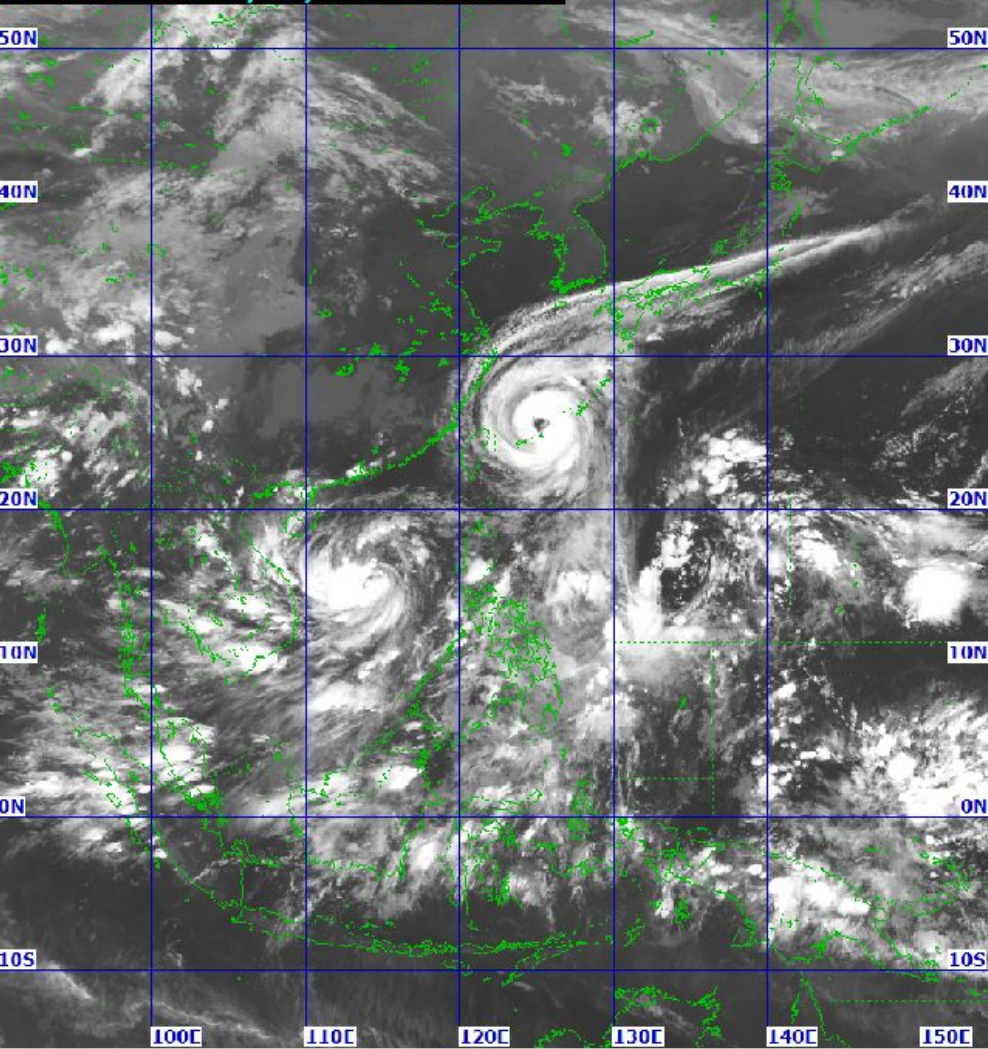
389 clicks



539 clicks



Himawari IR 13/09/2017 13:10UTC

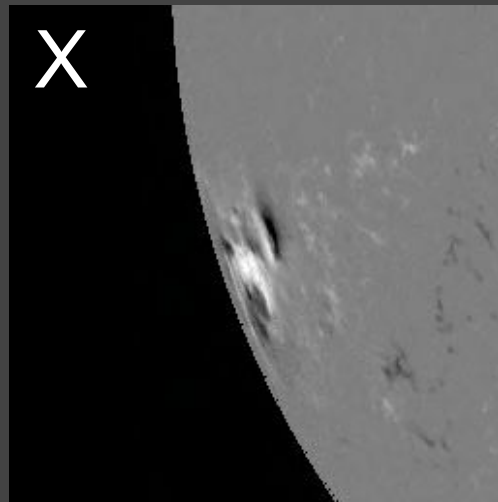
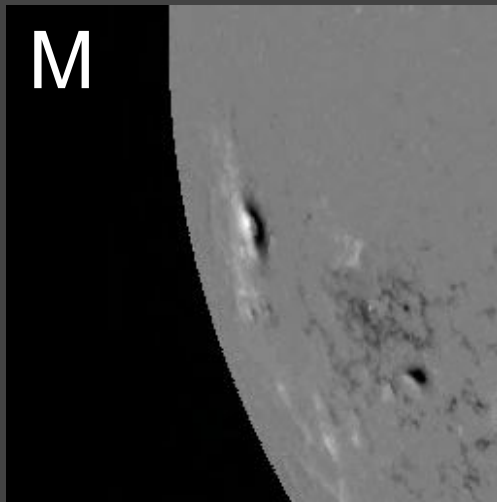
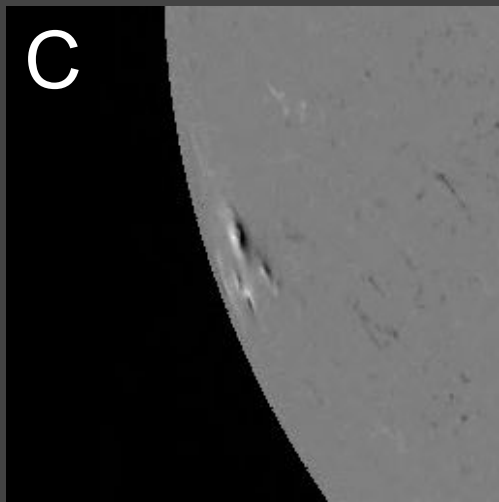
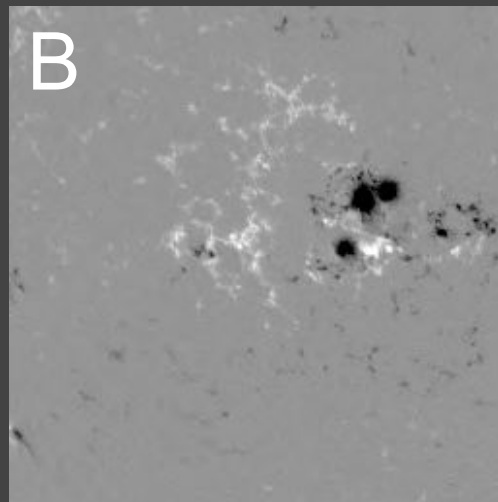
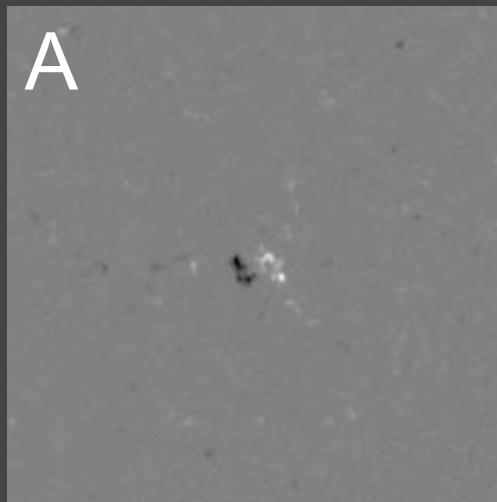
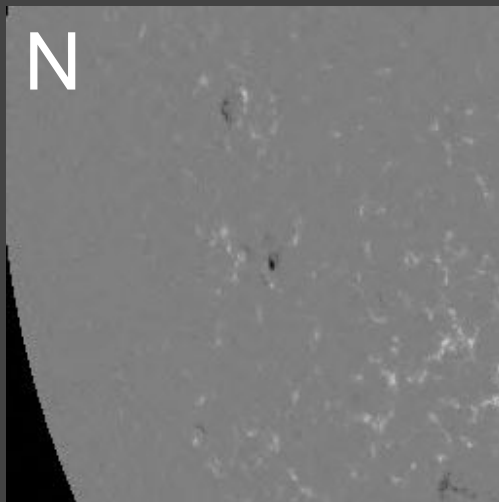








September 10, 2017 Solar X8.2 Flare Observed by SDO/AIA



Dataset

1996 ~

Negative (N/A/B)

train 111,245

val 12,603

test 9,200

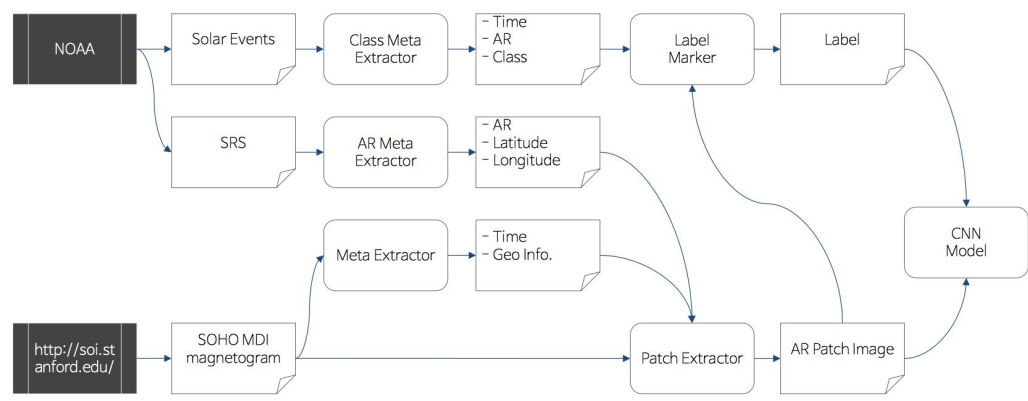
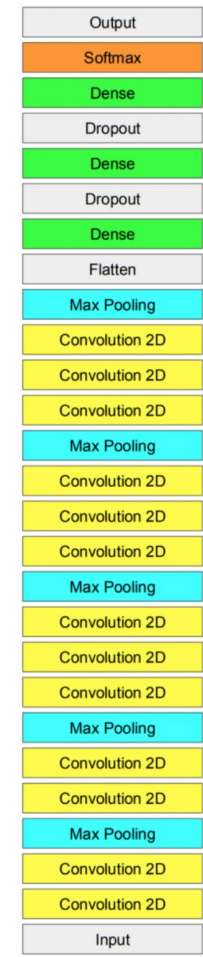
Positive (C/M/X)

train 140,710

val 16,852

test 6,595

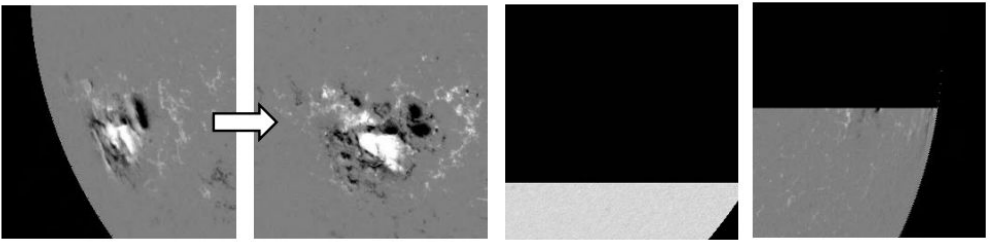
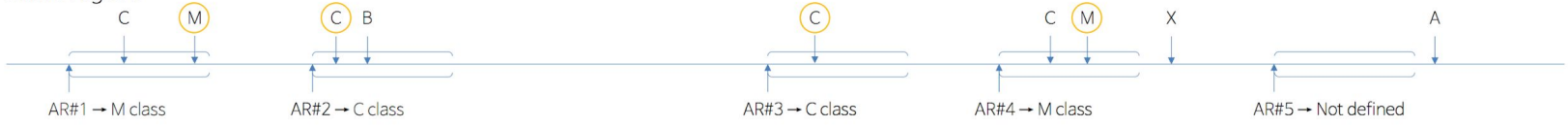
Classification : VGG-16



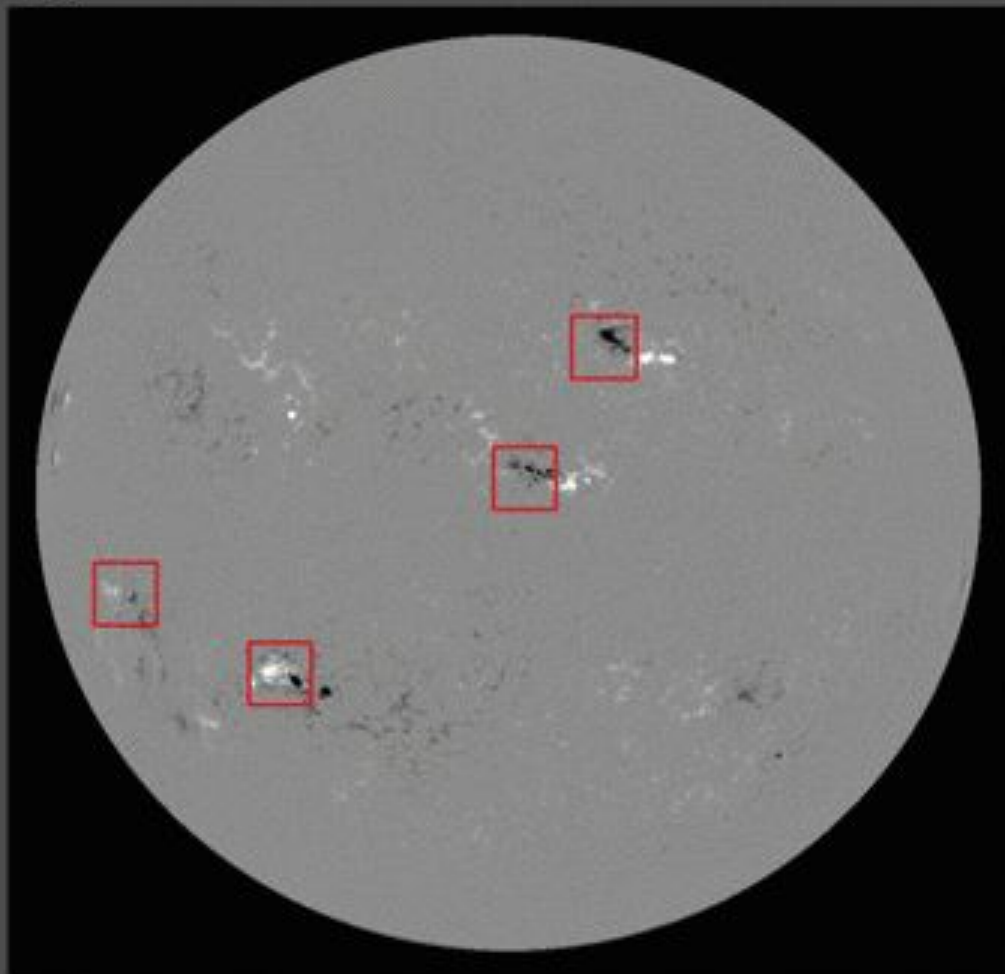
Binary	Class	Train	Validation	Test
Negative	Not defined	99,569	5,763	4,360
	A	56	0	0
	B	11,620	6,840	4,860
	SUM	111,245	12,603	9,220
Positive	C	89,473	9,799	5,192
	M	44,398	5,295	1,244
	X	6,839	1,758	159
	SUM	140,710	16,852	6,595

Forecast Mechanism

We implement a forecast mechanism for our model by matching every magnetogram image and maximum X-ray peak flux within 24 hour for a given active region.



열상표기



과열입력

과열입력

저장

입력파일목록

파일목록

2000_08_28_03_11_01_363_SOHO_...

2000_08_28_04_47_01_364_SOHO_...

2000_08_28_06_23_01_365_SOHO_...

+ 2000_08_28_07_59_01_366_SOHO_...

2000_08_28_09_35_01_367_SOHO_...

과열목록

과열목록

+

C

C

B

B

과열 수정

과열관리

추가

삭제

열대이동

이전

이후



커널

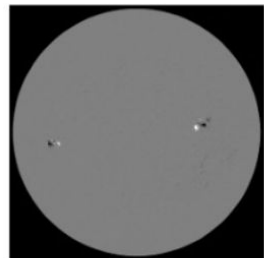
0.4

X

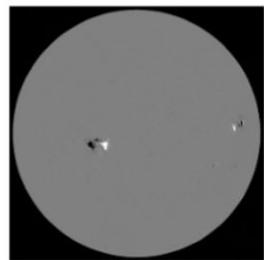
0.4

수정

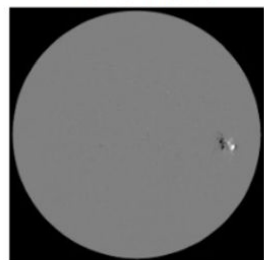
Input Image



1996_11_23_06_24_05_348_SOHO_MDI_MDI_magnetogram_origin.png

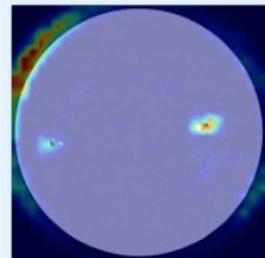


1996_11_25_00_03_05_373_SOHO_MDI_MDI_magnetogram_origin.png

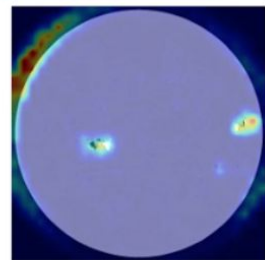


1996_11_29_20_48_05_439_SOHO_MDI_MDI_magnetogram_origin.png

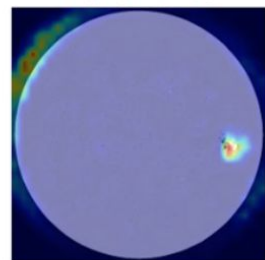
grad-cam



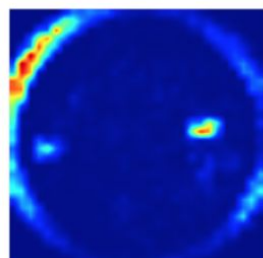
1996_11_23_06_24_05_348_SOHO_MDI_MDI_magnetogram_predic.png



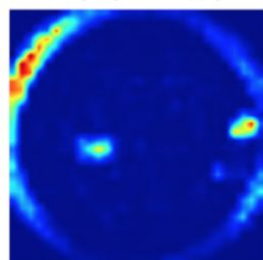
1996_11_25_00_03_05_373_SOHO_MDI_MDI_magnetogram_predic.png



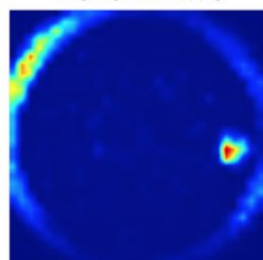
1996_11_29_20_48_05_439_SOHO_MDI_MDI_magnetogram_predic.png



1996_11_23_06_24_05_348_SOHO_MDI_MDI_magnetogram_heatmap.png

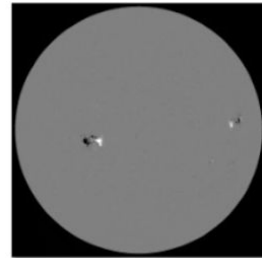


1996_11_25_00_03_05_373_SOHO_MDI_MDI_magnetogram_heatmap.png

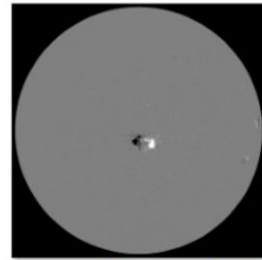


1996_11_29_20_48_05_439_SOHO_MDI_MDI_magnetogram_heatmap.png

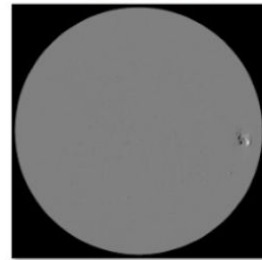
Input Image



1996_11_24_17_36_05_369_SOHO_MDI_MDI_magnetogram_origin.png

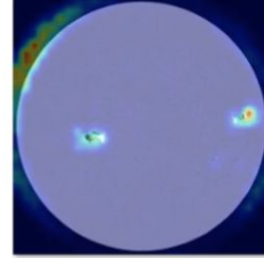


1996_11_26_12_51_05_395_SOHO_MDI_MDI_magnetogram_origin.png

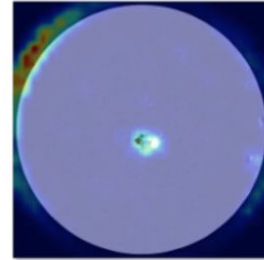


1996_11_30_17_39_05_449_SOHO_MDI_MDI_magnetogram_origin.png

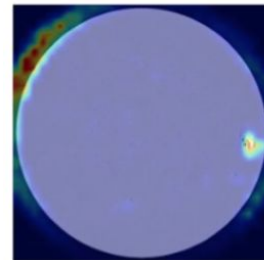
grad-cam



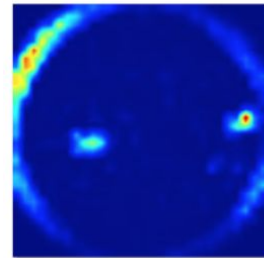
1996_11_24_17_36_05_369_SOHO_MDI_MDI_magnetogram_predic.png



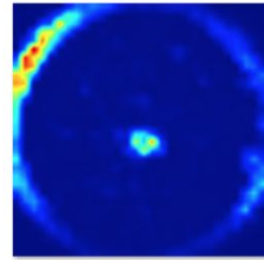
1996_11_26_12_51_05_395_SOHO_MDI_MDI_magnetogram_predic.png



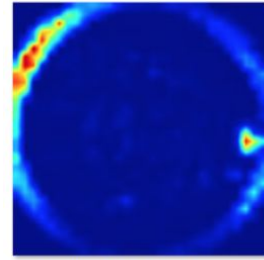
1996_11_30_17_39_05_449_SOHO_MDI_MDI_magnetogram_predic.png



1996_11_24_17_36_05_369_SOHO_MDI_MDI_magnetogram_heatmap.png

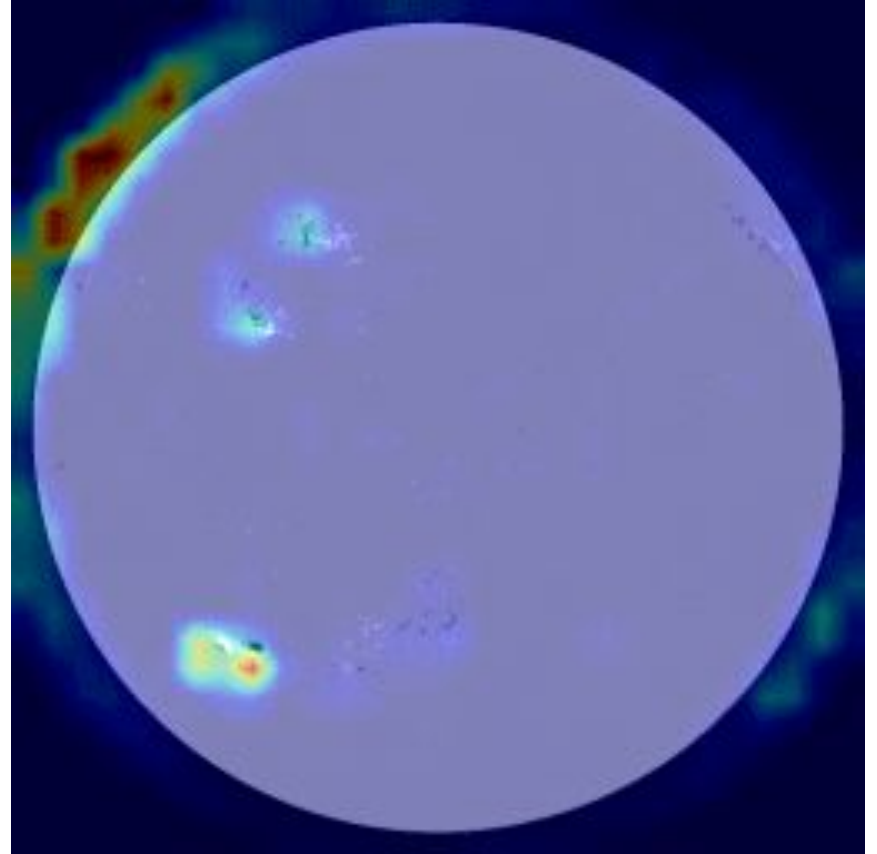
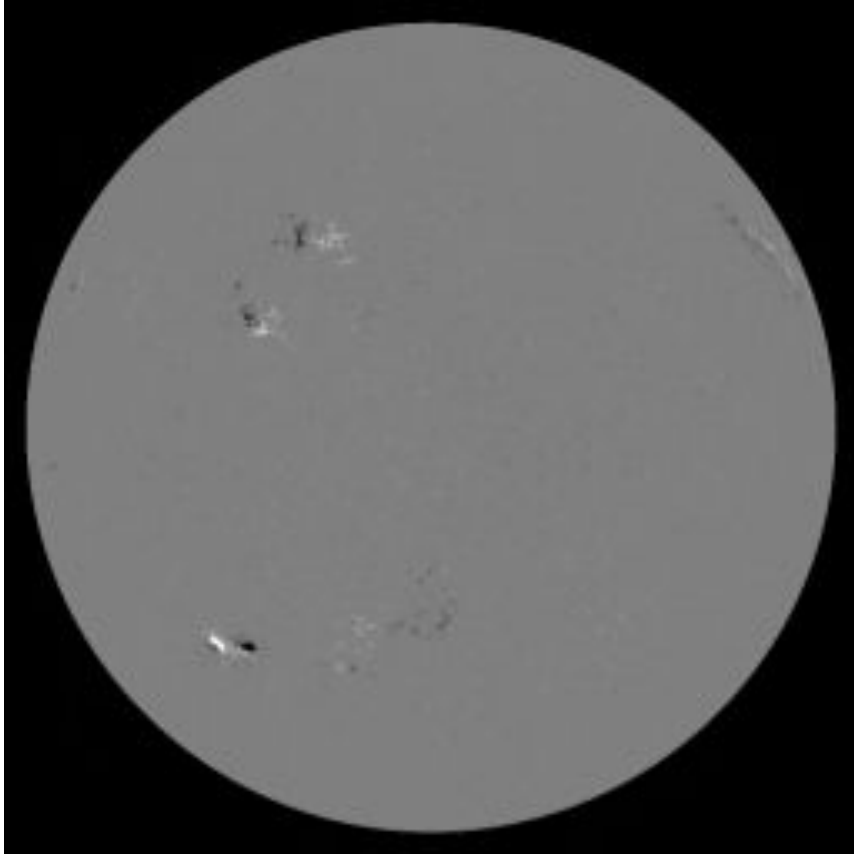


1996_11_26_12_51_05_395_SOHO_MDI_MDI_magnetogram_heatmap.png

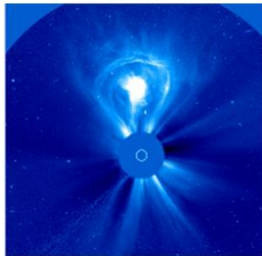
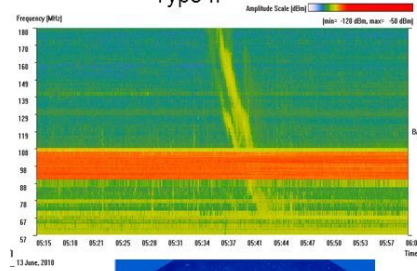


1996_11_30_17_39_05_449_SOHO_MDI_MDI_magnetogram_heatmap.png

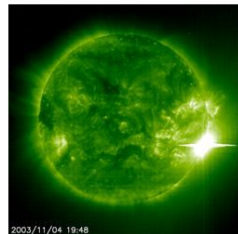
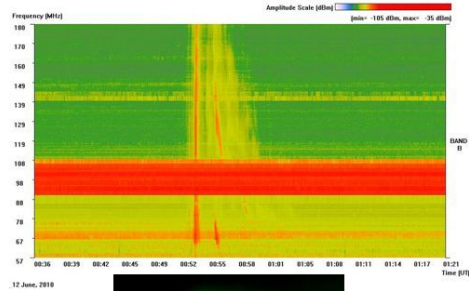
Magnetogram C, M, X Class Gif



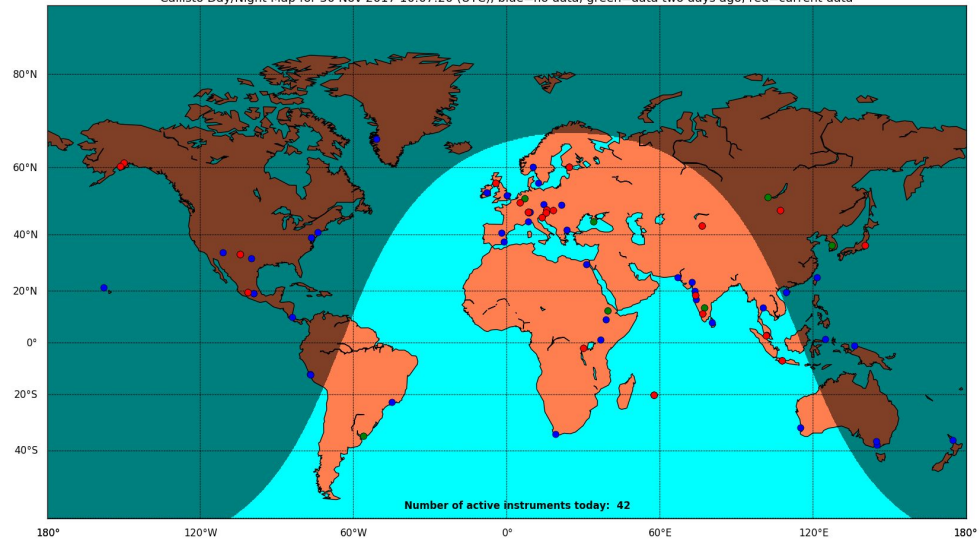
Type II



Type III

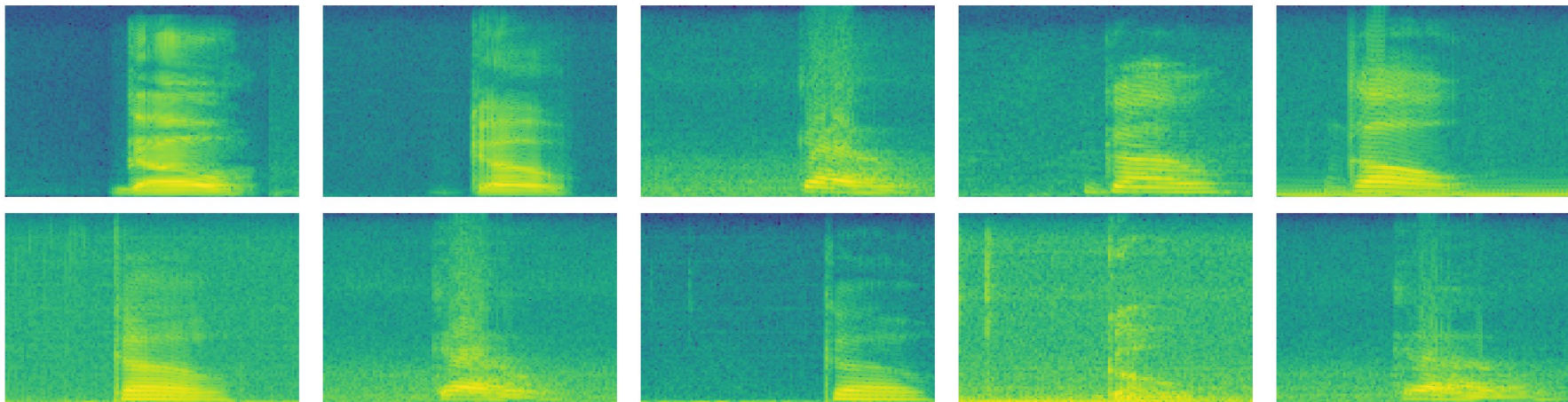


Callisto Day/Night Map for 30 Nov 2017 10:07:20 (UTC), blue=no data, green=data two days ago, red=current data

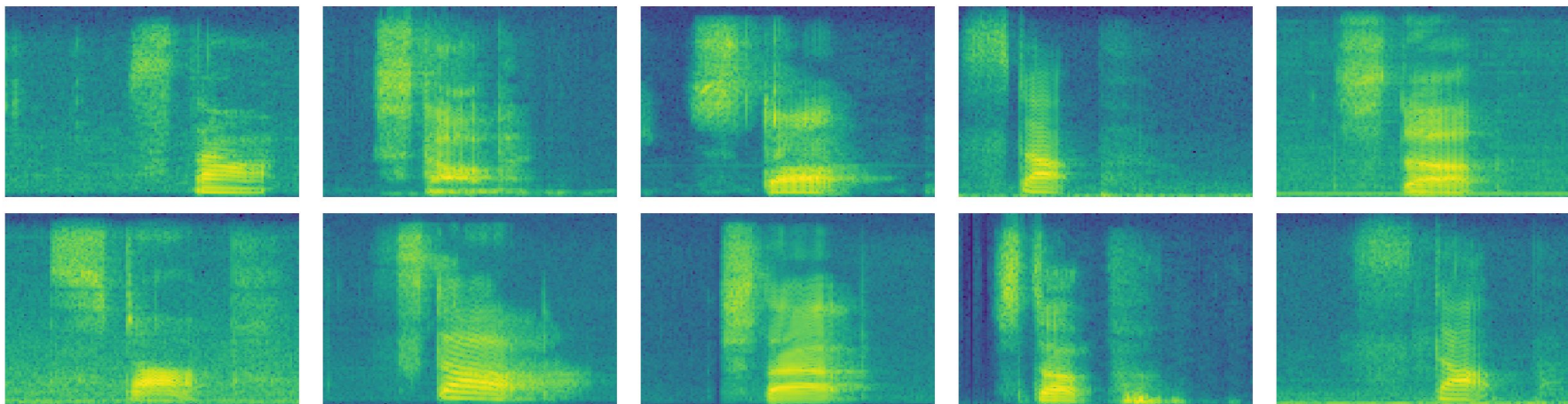


TensorFlow Speech Recognition Challenge

GO

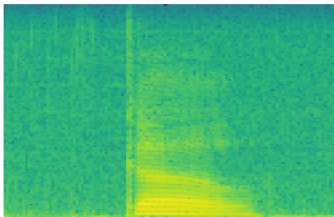
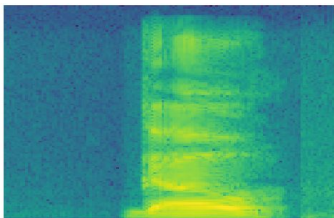


STOP



TensorFlow Speech Recognition Challenge

GO



2. 모델 구성하기

```
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(256, 256, 3)))
model.add(Conv2D(32, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
```

```
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
```

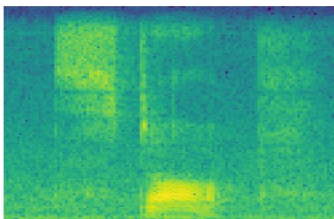
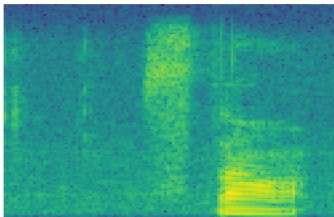
```
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(2, activation='softmax'))
```

3. 모델 학습과정 설정하기

```
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
Epoch 11/15
46/46 [=====] - 500s - loss: 0.1304 - acc: 0.9484 - val_loss: 0.1726 - val_acc: 0.9259
Epoch 12/15
46/46 [=====] - 506s - loss: 0.1416 - acc: 0.9382 - val_loss: 0.1636 - val_acc: 0.9468
Epoch 13/15
46/46 [=====] - 475s - loss: 0.1228 - acc: 0.9436 - val_loss: 0.1997 - val_acc: 0.9444
Epoch 14/15
46/46 [=====] - 501s - loss: 0.1480 - acc: 0.9341 - val_loss: 0.1869 - val_acc: 0.9213
Epoch 15/15
46/46 [=====] - 501s - loss: 0.1445 - acc: 0.9368 - val_loss: 0.1693 - val_acc: 0.9306
-- Evaluate --
acc: 92.59%
```

STOP



정답



정답



학습



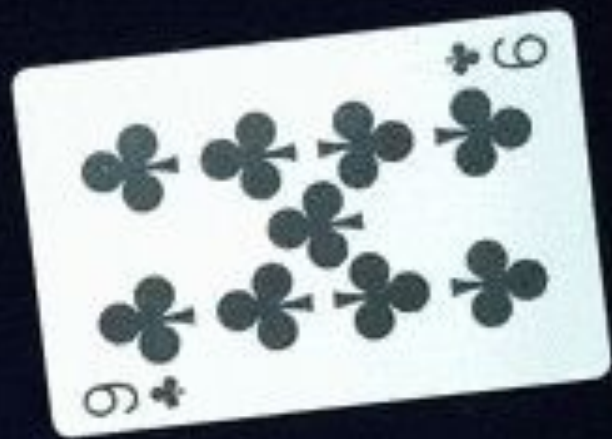
입력



모델



출력







History

IMG_6170.jpg

- Open
- Paste
- Rotate Canvas
- Master Opacity Change
- Invert
- Nudge

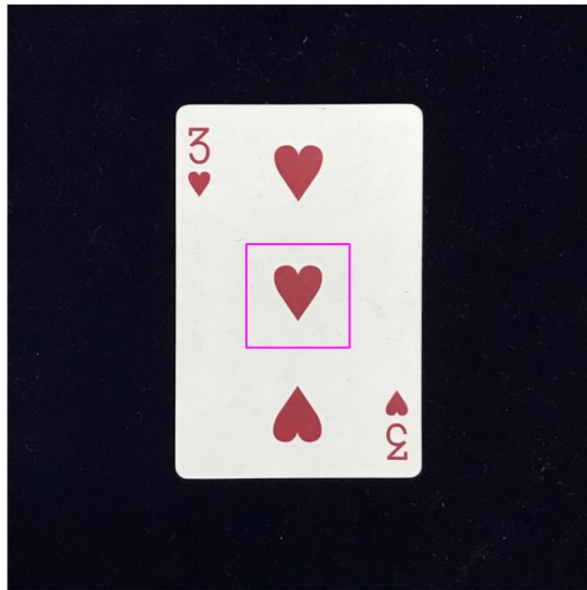
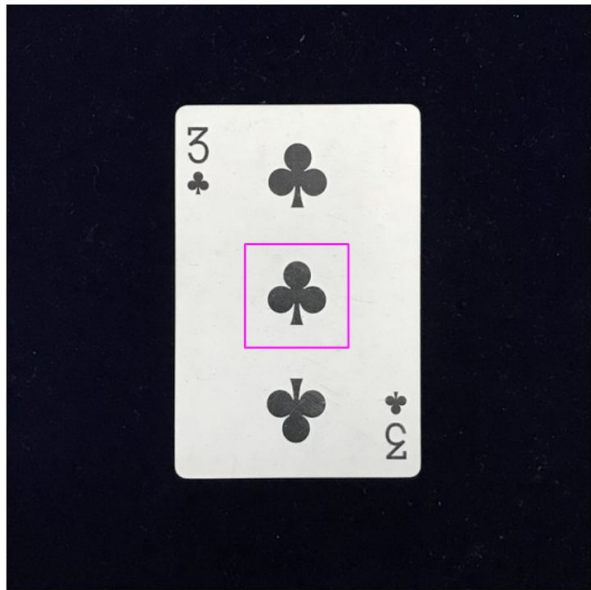
Layers Channels Paths

Kind

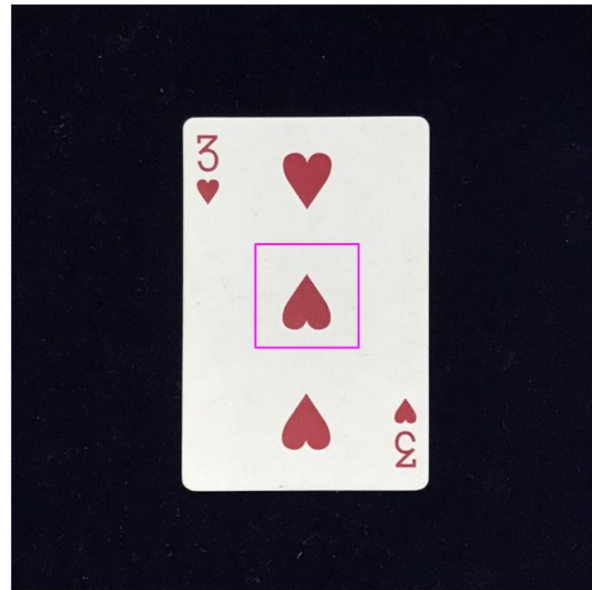
Normal Opacity: 50%

Lock: Fill: 100%

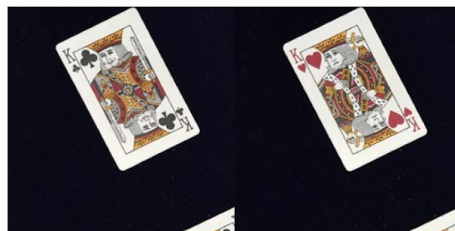
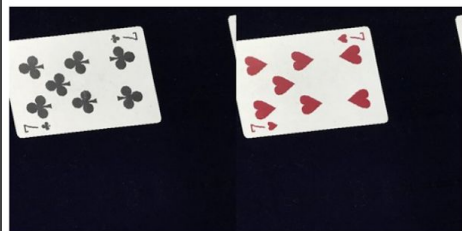
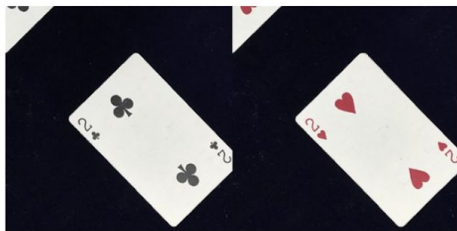
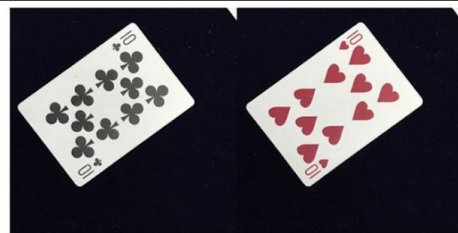
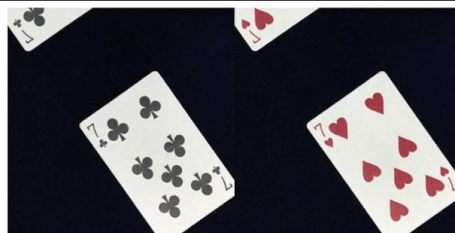
- Layer 1
- Background

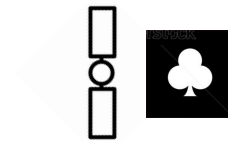
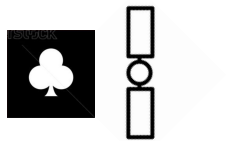


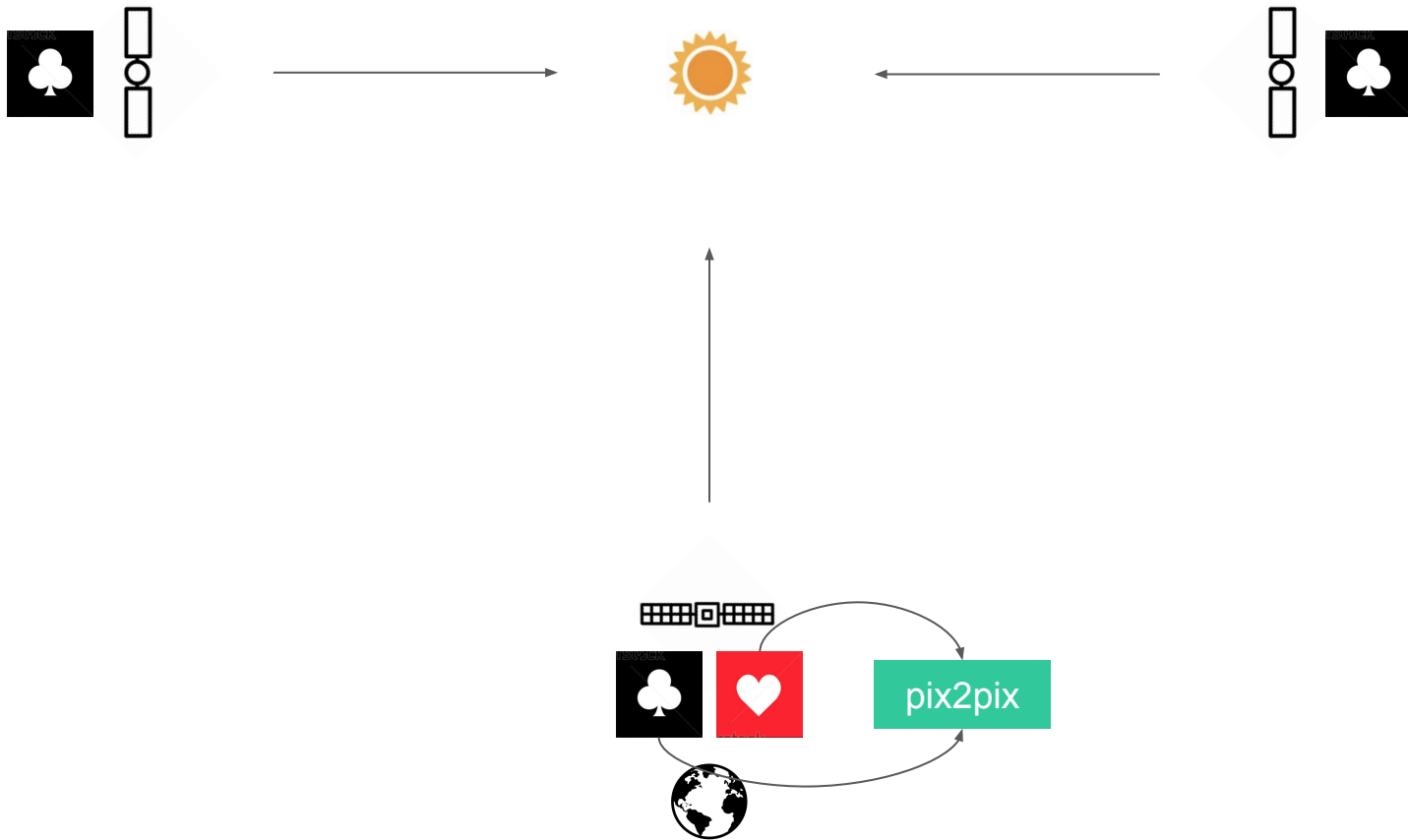
O

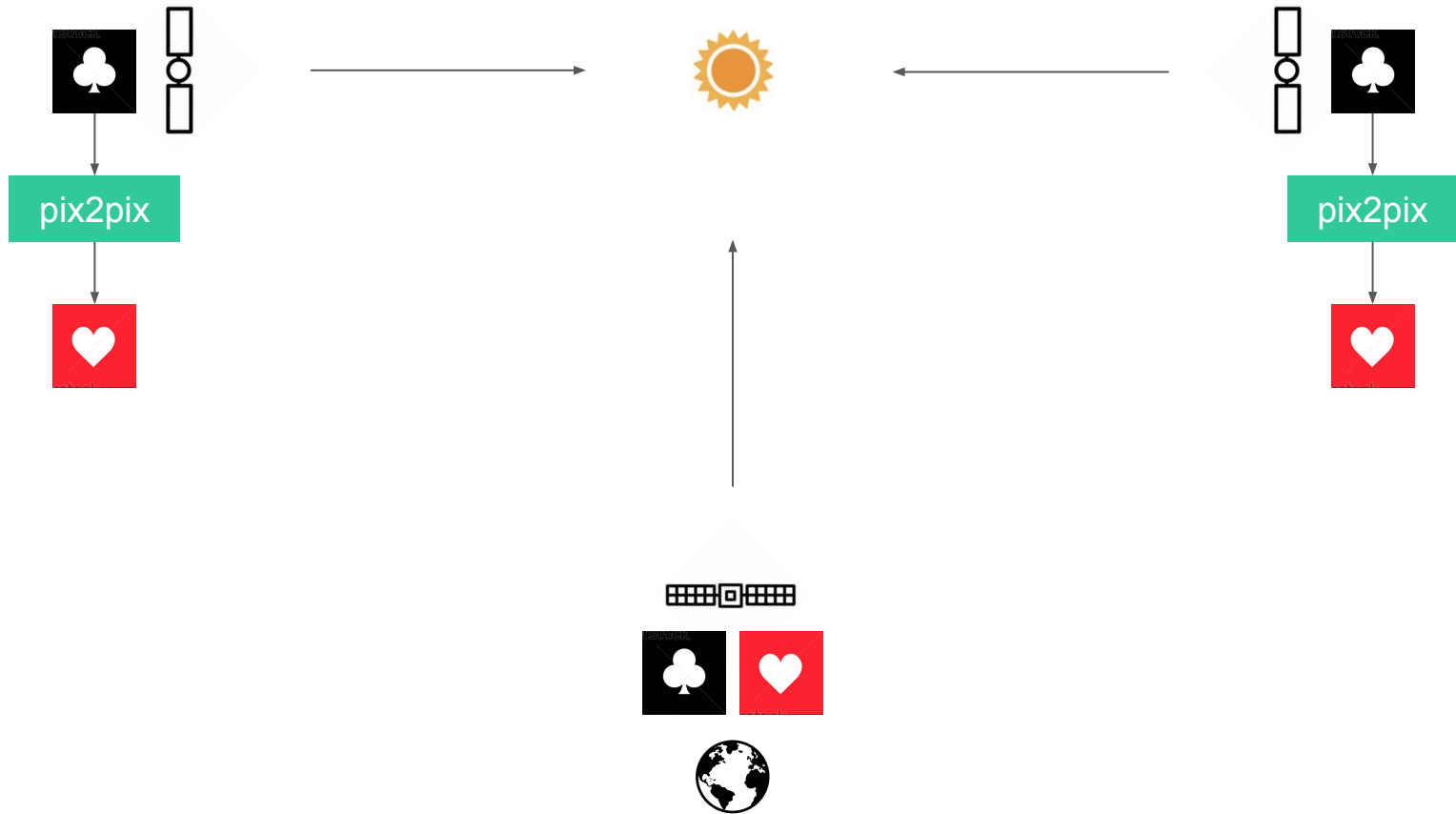


X

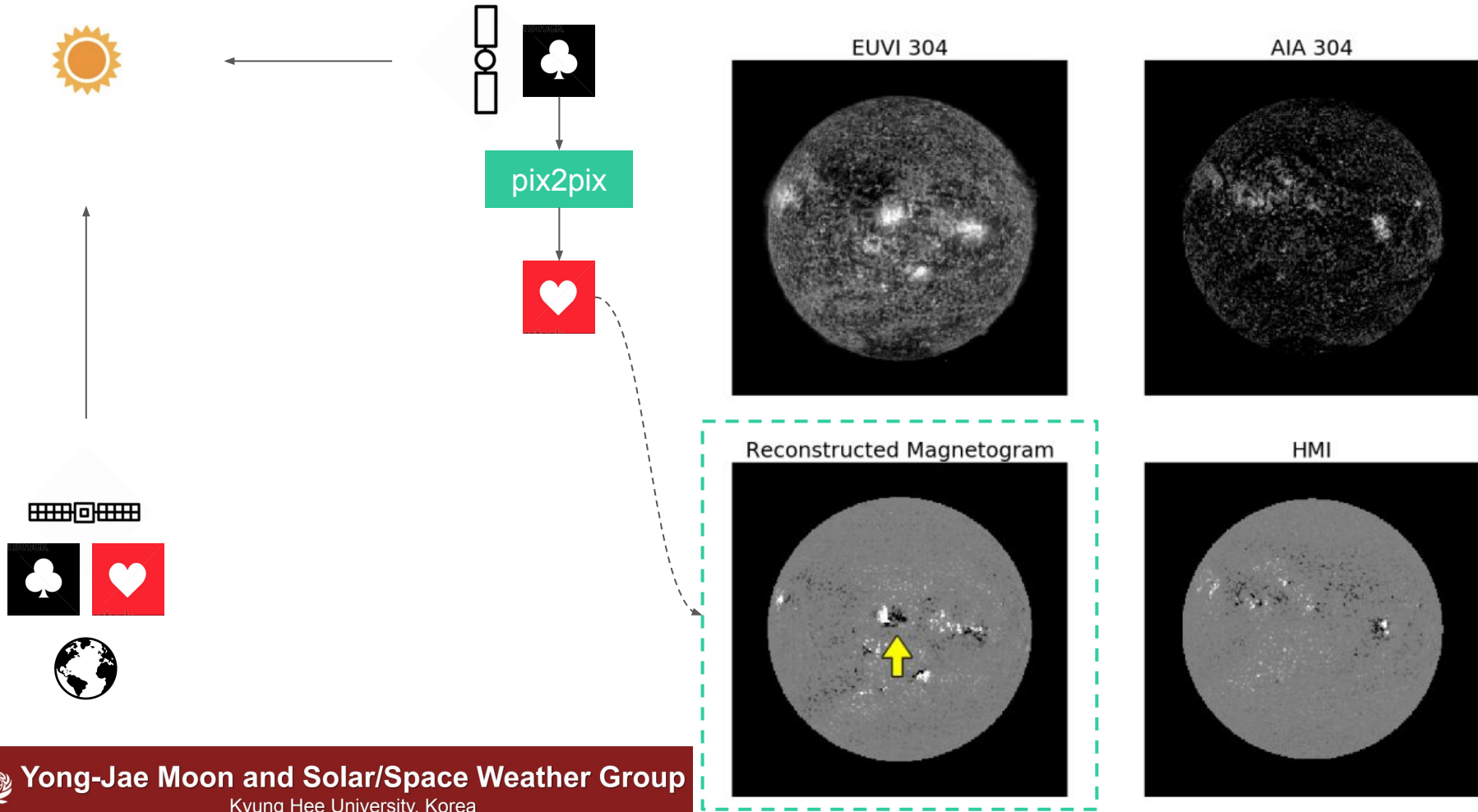


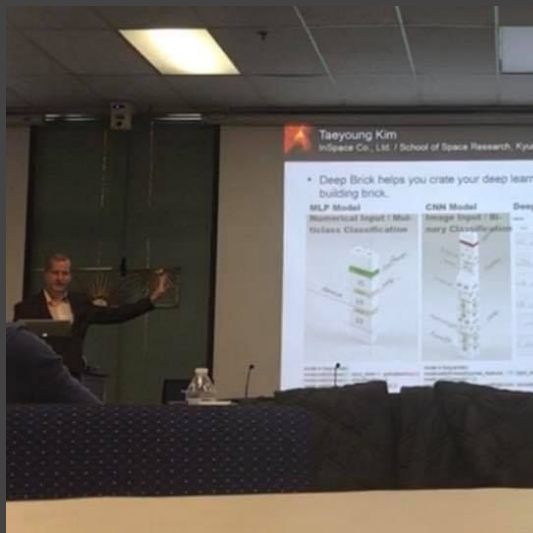
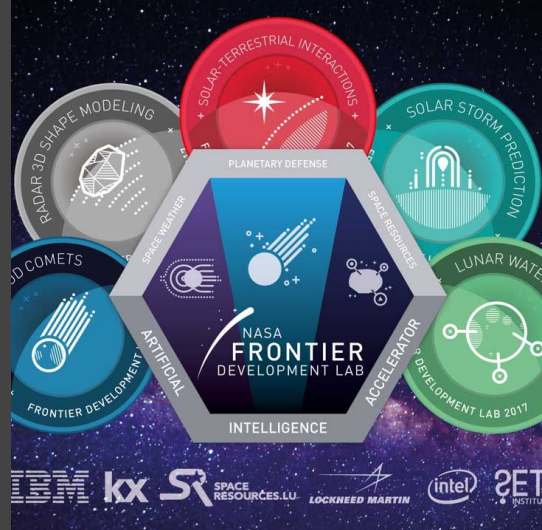






2016_04_02_00

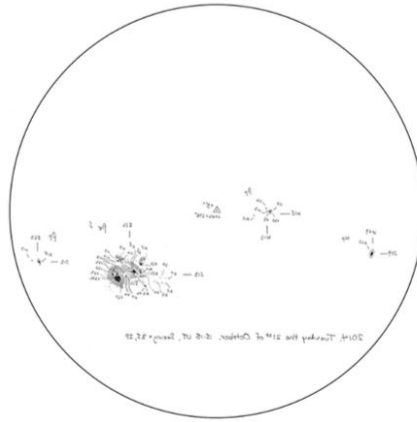




Next...

Purpose

Sunspot Drawing

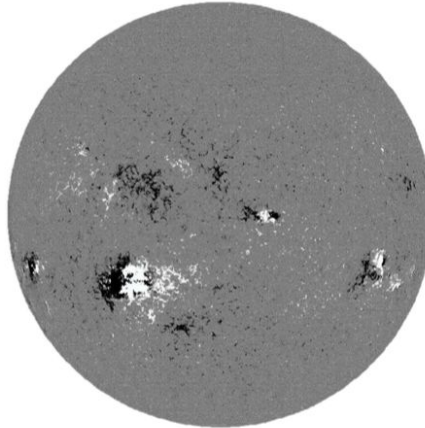


1600

1800

2000

Space Weather Quantities

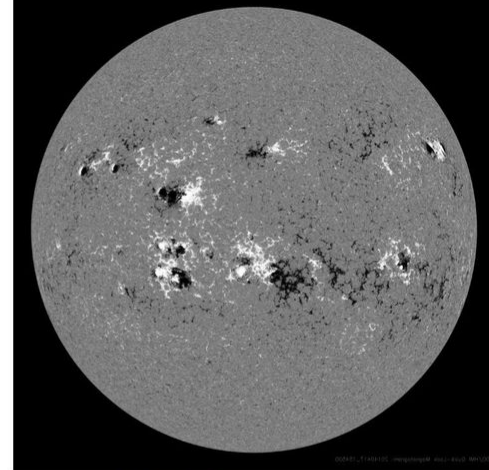
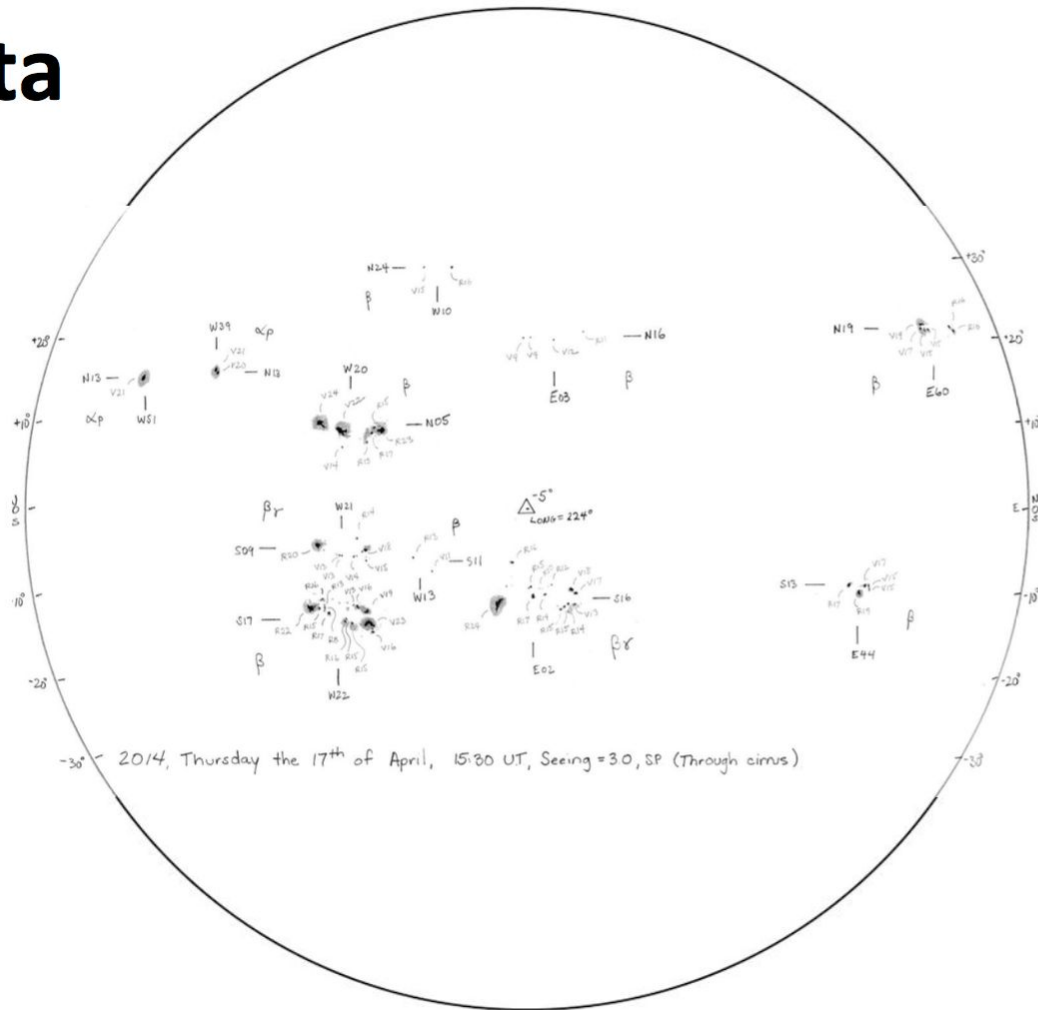


Magnetogram

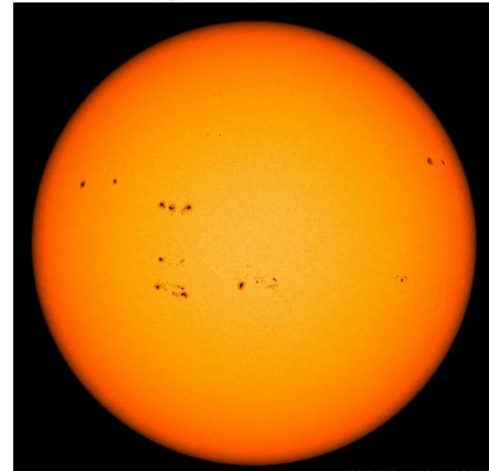
1900

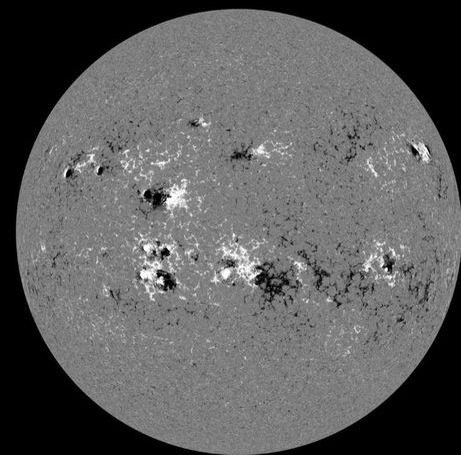
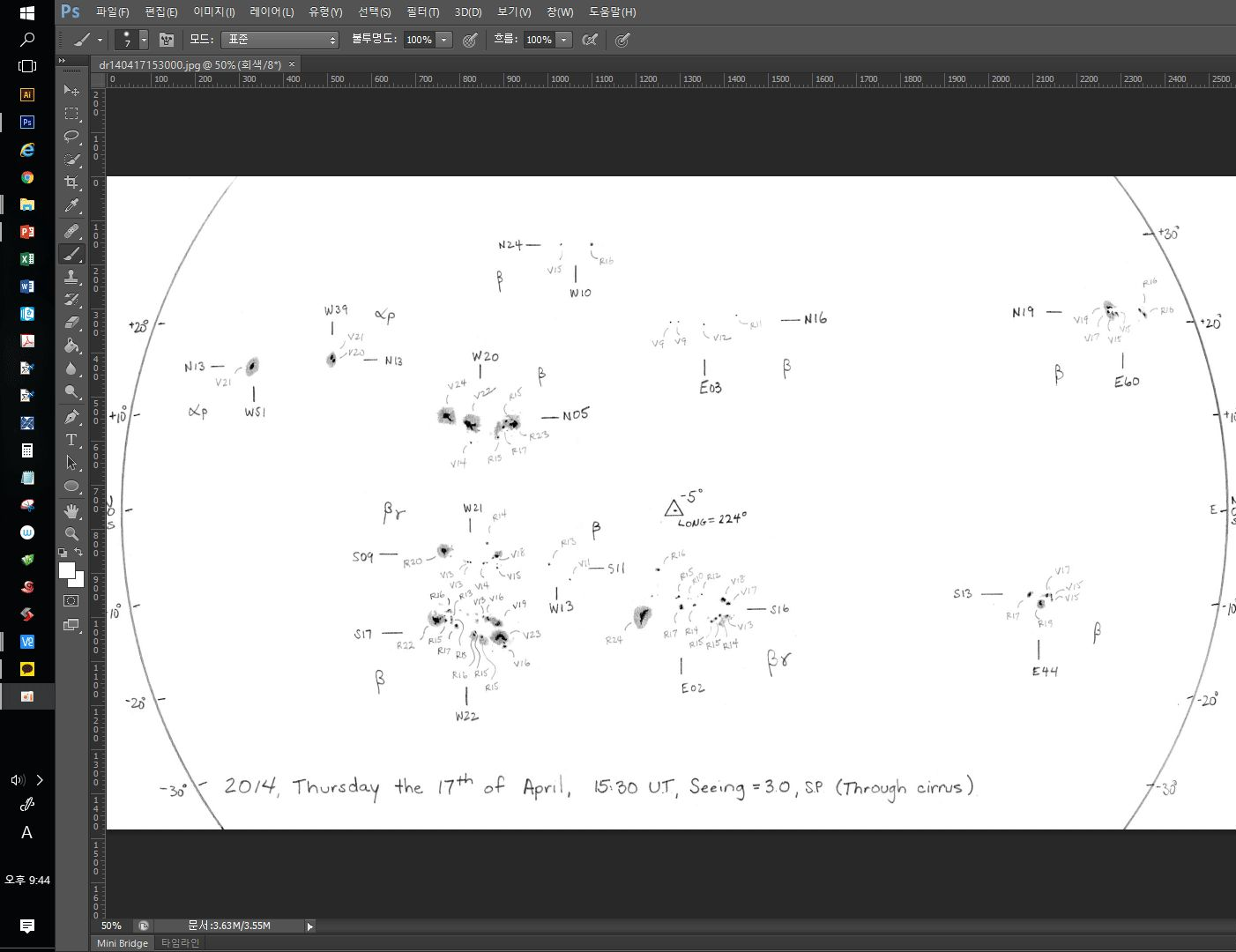
2000

Data

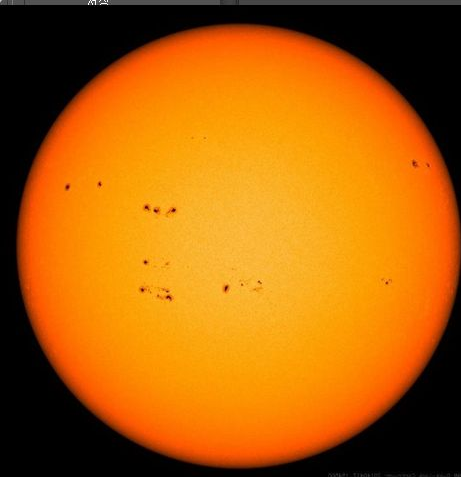


▲ SDO/HMI magnetogram
2014.04.17 15:30:00 [UT]
▼ SDO/HMI Continuum

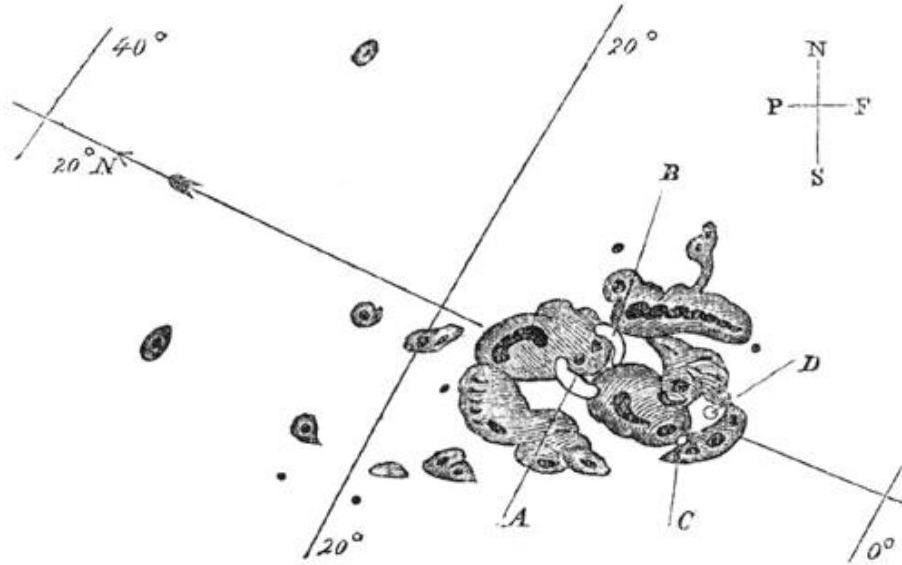




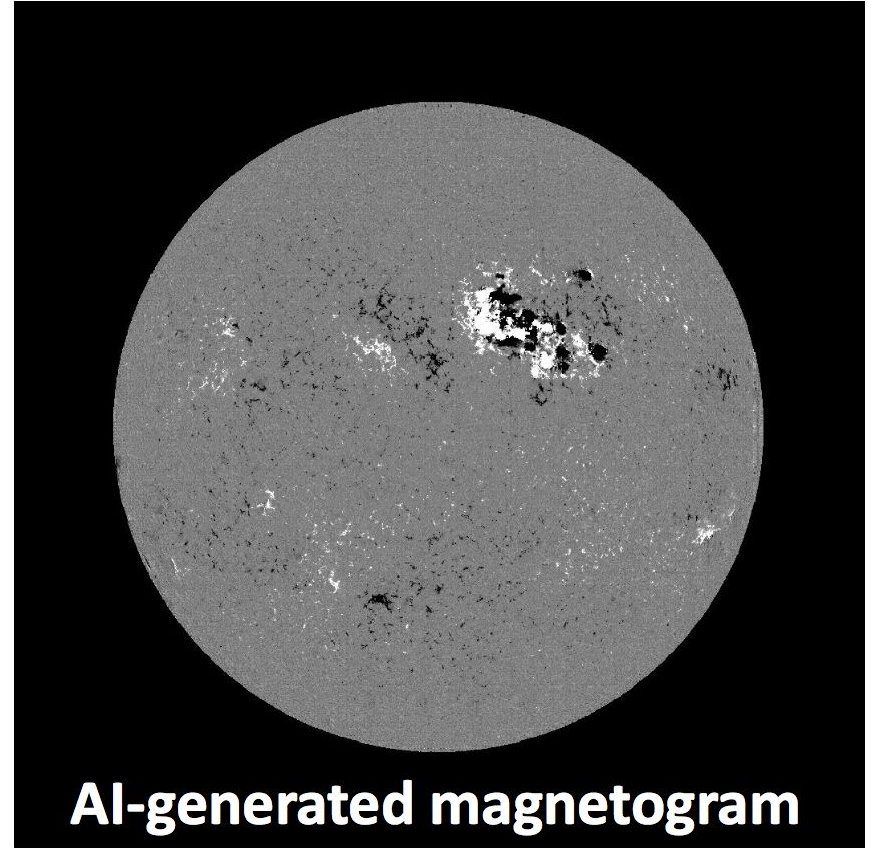
확대 0.5% F3
축소 0.5% F4
클리어 마스크 F5
클리어 마스크 적용 F6
반전 png 저장 F7



Generation: the Carrington sunspot!

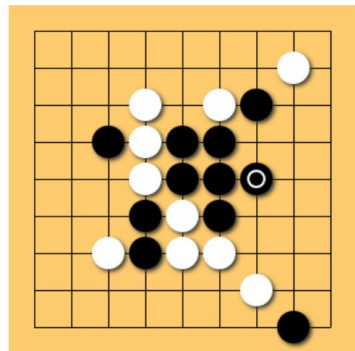


Carrington sunspot in 1859



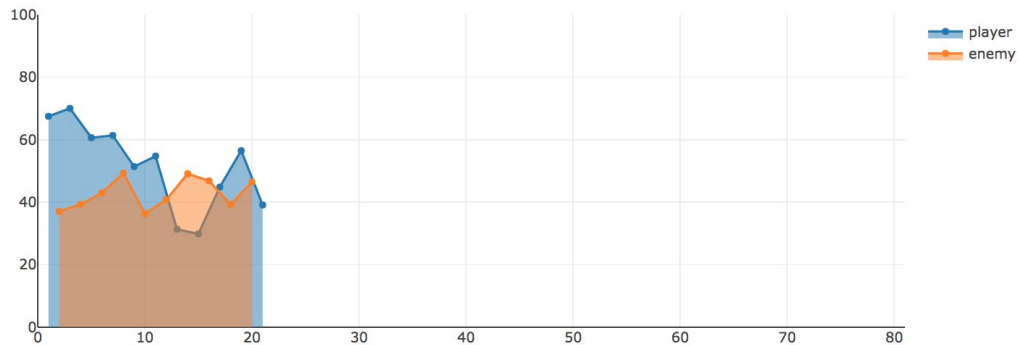
AI-generated magnetogram

AlphaOmoc vs AlphaOmoc

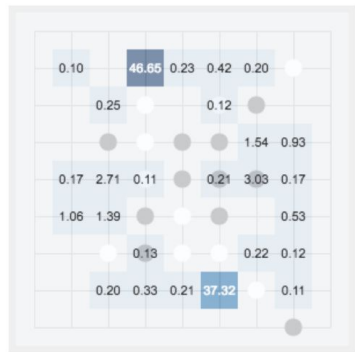


Player : simulation: 500
Enemy : simulation: 375

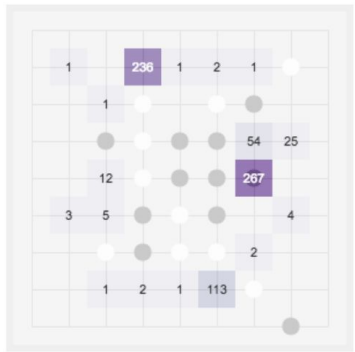
Value Network



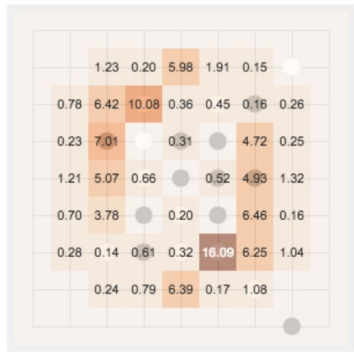
Player Policy Network



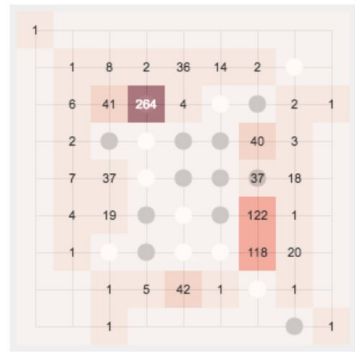
Player MCTS Visit



Enemy Policy Network



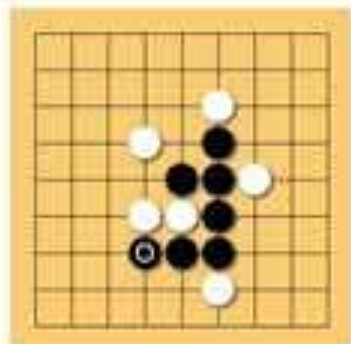
Enemy MCTS Visit



훈
수
+
감
정



Player: Entity:



simulation: 400

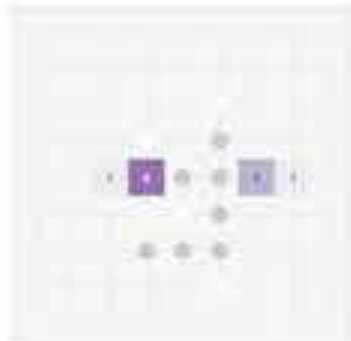
Value Network



Player Policy Network



Player MCTS Visit

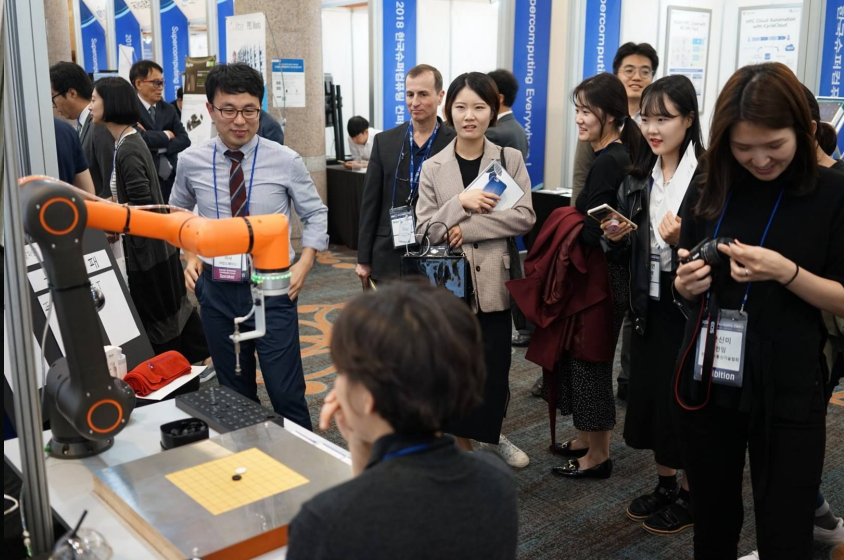


Enemy MCTS Visit



Enemy Policy Network

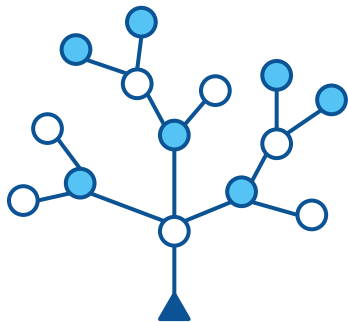




Deep Tree

감성

병렬
분산
처리



알파
오목

로봇암

비전



임근영 ○ GeunYong
C++, Python3, TeX



민규식 ○ Kyusik



송규예 ● Rachel



Youngha Kim



김정대 ○



김준태 ○ 김준태



김태영 ●



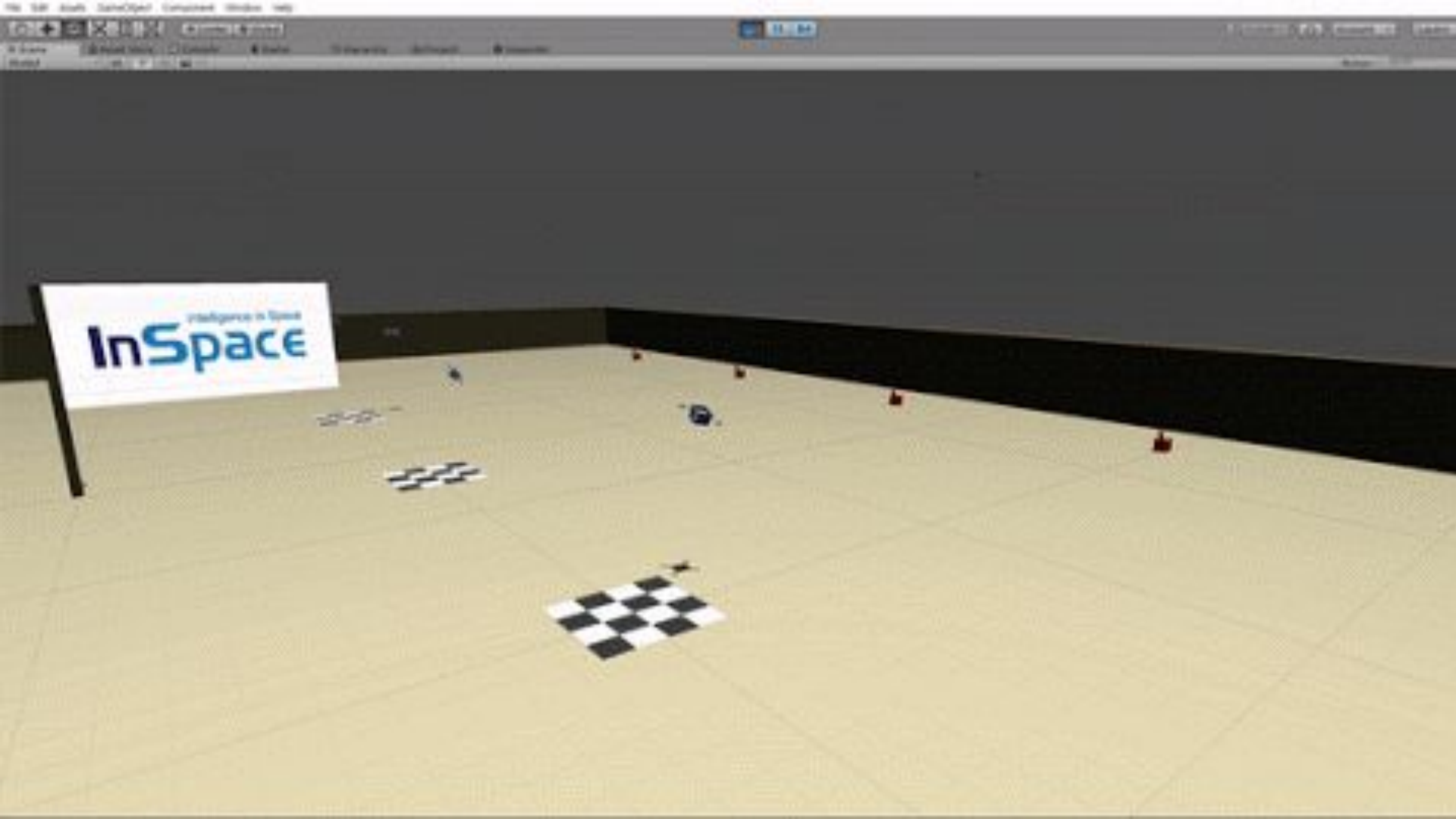
박세민 ●



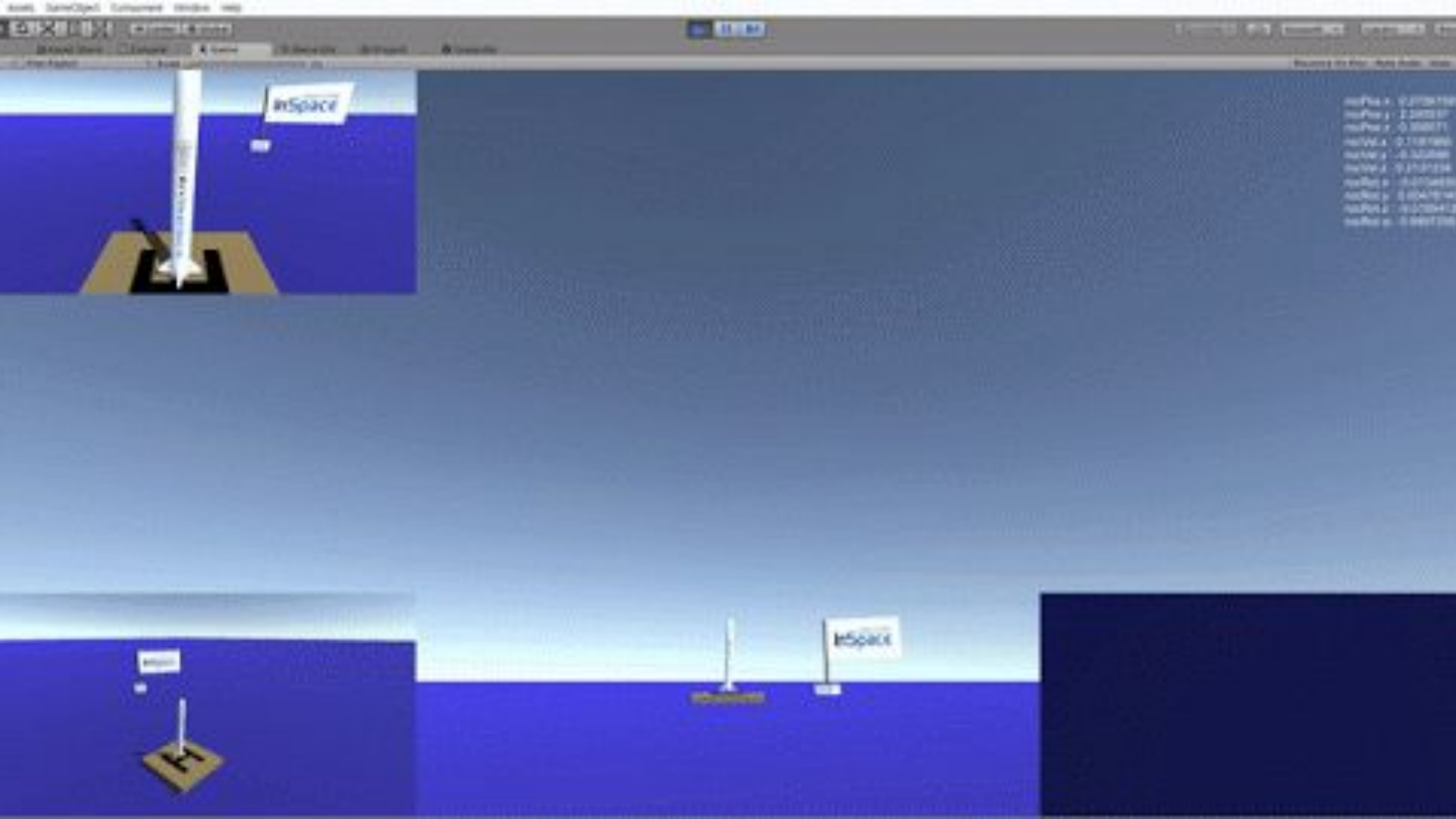
이현호 ○



조양규 ○ 조양규



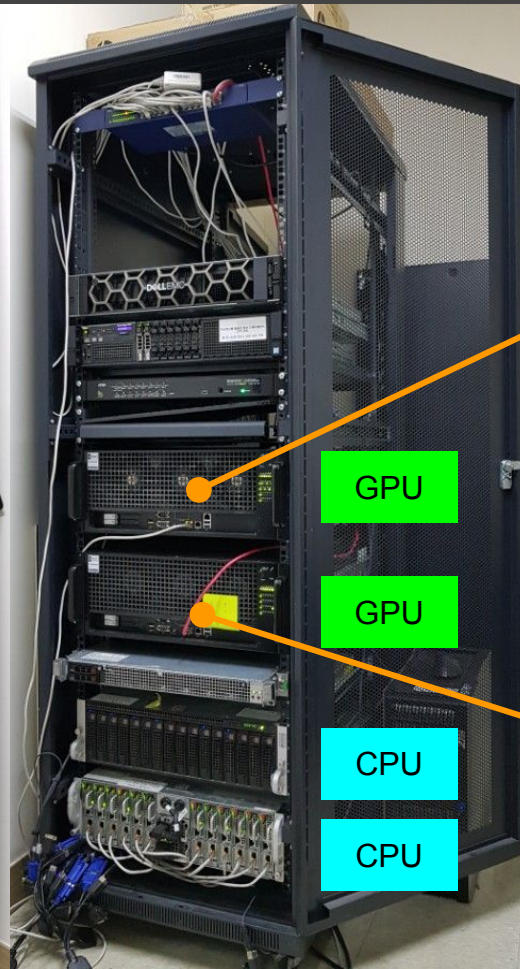








CPU



GPU

GPU

CPU

CPU

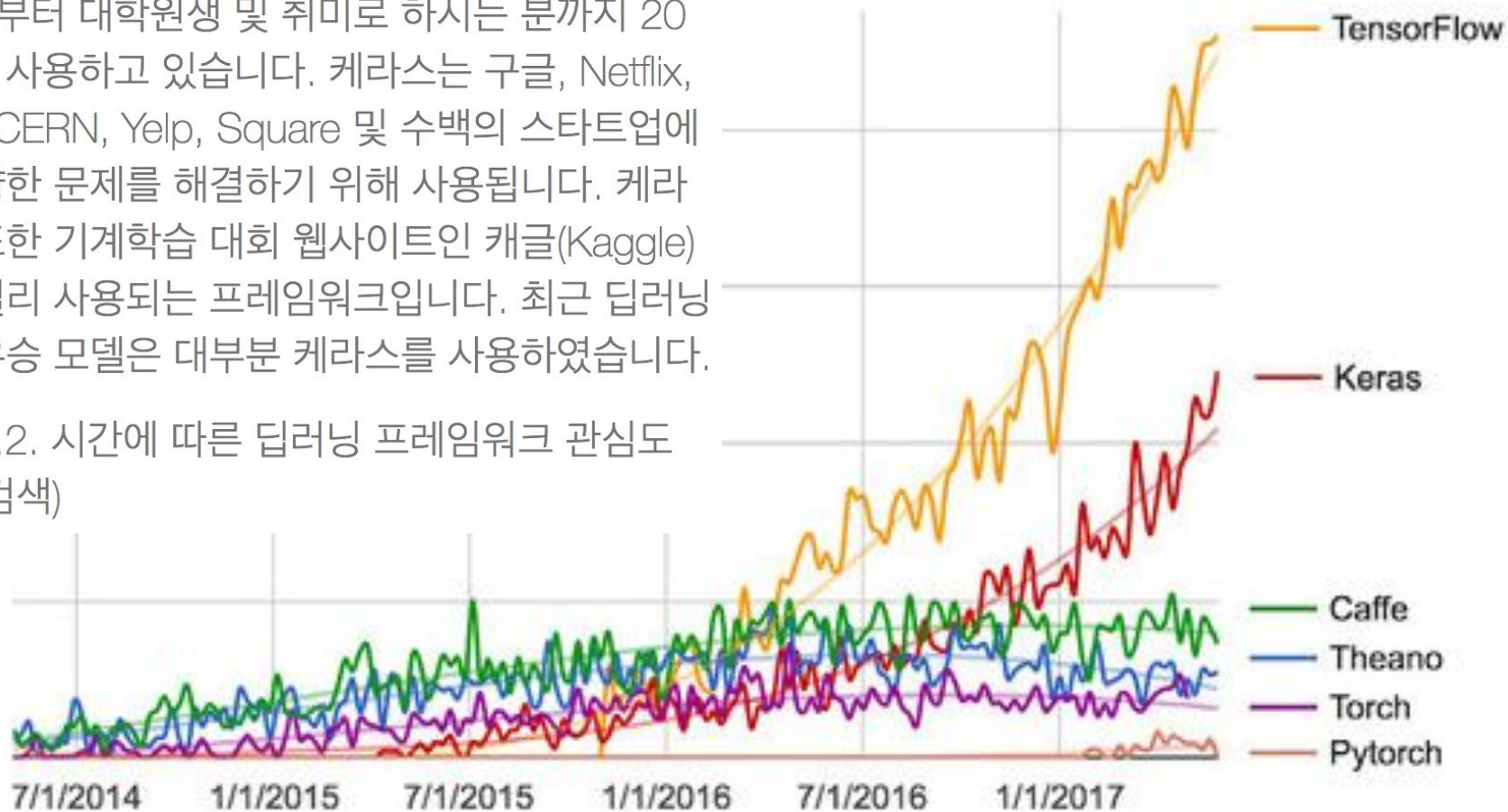


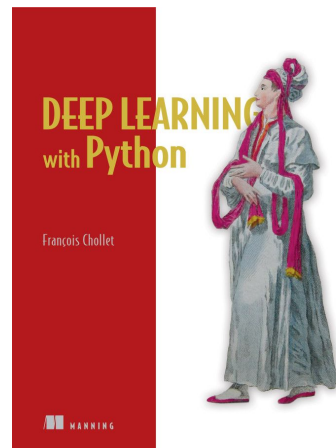
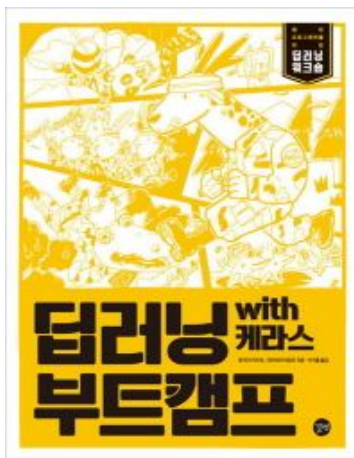
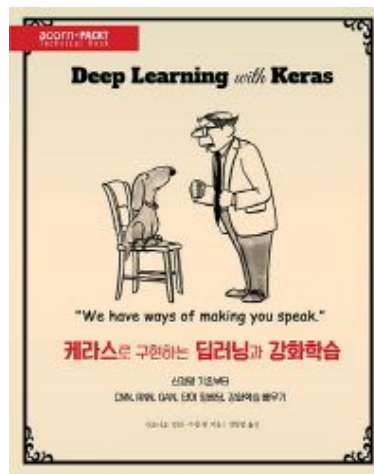
케라스 이야기



케라스는 스타트업 및 대기업의 학술 연구자 및 엔지니어부터 대학원생 및 취미로 하시는 분까지 20만명이 사용하고 있습니다. 케라스는 구글, Netflix, Uber, CERN, Yelp, Square 및 수백의 스타트업에서 다양한 문제를 해결하기 위해 사용됩니다. 케라스는 또한 기계학습 대회 웹사이트인 캐글(Kaggle)에서 널리 사용되는 프레임워크입니다. 최근 딥러닝 대회 우승 모델은 대부분 케라스를 사용하였습니다.

그림 3.2. 시간에 따른 딥러닝 프레임워크 관심도 (구글 검색)





- **하용호**님, 카카오 데이터사이언티스트 - 뜯구름같은 딥러닝 이론을 블록이라는 손에 잡히는 실체로 만져가며 알 수 있게 하고, 구현의 어려움은 케라스라는 시를 읽듯이 읽어내려 갈 수 있는 라이브러리로 풀어준다.
- **이부일**님, (주)인사아트마이닝 대표 - 여행에서도 좋은 가이드가 있으면 여행지에 대한 깊은 이해로 여행이 풍성해지듯이 이 책은 딥러닝이라는 분야를 여행할 사람들에게 가장 훌륭한 가이드가 되리라고 자부할 수 있다. 이 책을 통하여 딥러닝에 대해 보지 못했던 것들이 보이고, 듣지 못했던 것들이 들리고, 말하지 못했던 것들이 말해지는 경험을 하게 될 것이다.
- **이활석**님, 네이버 클로바팀 - 레고 블록에 비유하여 누구나 이해할 수 있게 쉽게 설명해 놓은 이 책은 딥러닝의 입문 도서로서 제 역할을 다 하리라 믿습니다.
- **김진중**님, 야놀자 Head of STL - 복잡했던 머릿속이 맑고 깨끗해지는 효과가 있습니다.
- **이태영**님, 신한은행 디지털 전략부 AI LAB - 기존의 텐서플로우를 활용했던 분들에게 바라볼 수 있는 관점의 전환점을 줄 수 있는 Mild Stone과 같은 책이다.
- **전태균**님, 썬트렉아이 - 케라스의 특징인 단순함, 확장성, 재사용성을 눈으로 쉽게 보여주기 위해 친절하게 정리된 내용이라 생각합니다.
- **유재준**님, 카이스트 - 바로 적용해보고 싶지만 어디부터 시작할지 모를 때 최선의 선택입니다.



케라스 코리아 (Keras Korea)

공개 그룹

정보

토론

멤버

이벤트

동영상

사진

그룹 인사이트

그룹 관리

이 그룹 검색



바로가기



대답사



가입함 ▼

✓ 알림

공유하기

... 더 보기



게시물 작성



사진/동영상 추가



라이브 방송



더 보기



글쓰기...

멤버 추가

+ 이름이나 이메일 주소를 입력하세요...

멤버

케라스 코리아에서는 **각 분야의 전문가**들이 딥러닝 기술을 쉽게 접목하거나 딥러닝 기술에 대해 입문하시고 싶은 분들에게 도움을 드리고 소통하는 것에 목표를 두고 있습니다.

모두의 손에 딥러닝 툴을...

THANK YOU

SOSCON 2018

SAMSUNG OPEN SOURCE CONFERENCE 2018

