

Human pose estimation 101 with Tensorflow



Jaewook Kang, Ph.D.

jwkang10@gmail.com

Oct. 2018

소 개



Jaewook Kang (강재
욱)

GIST EEC Ph.D. (2015)
 soundly 연구팀 리더 (~2018.5)

 MoTLab Director (2018.1~)

 Jeju DL Camp (2018.6-7)

 Google Developers Experts (2018, 10~)

좋아하는 것:

통계적 신호처리 / Compressed sensing

C++ Native Audio DSP 라이브러리 구현

Mobile Machine learning

Pose Estimation / Computer Vision

Tensorflow / Cloud TPU

수영 덕력 6년



Human Pose Estimation



- ❖ Human Pose Estimation을 한마디로 요약하면?
 - 인물 카메라 입력으로부터 신체구조를 찾는 것



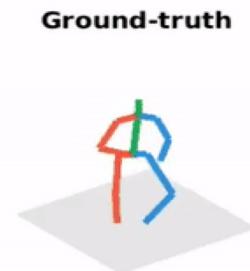
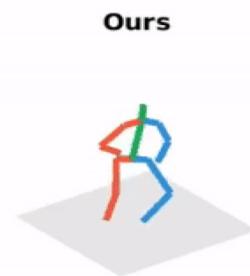
Human Pose Estimation



- ❖ Human Pose Estimation을 한마디로 요약하면?
 - 인물 카메라 입력으로부터 신체구조를 찾는 것



Eating



3D Human Pose Estimation in the Wild by Adversarial Learning, CVPR2018
Wei Yang, [Wanli Ouyang](#), [Xiaolong Wang](#), [Jimmy Ren](#), [Hongsheng Li](#), [Xiaogang Wang](#)

누구나 TensorFlow!

J. Kang Ph.D.

Human Pose Estimation



- 오늘은 pose estimation의 가장 기본이 되는

2D single pose estimation

- ~~졸라만 에스티멘이션~~ 만 다릅니다!



Human Pose Estimation

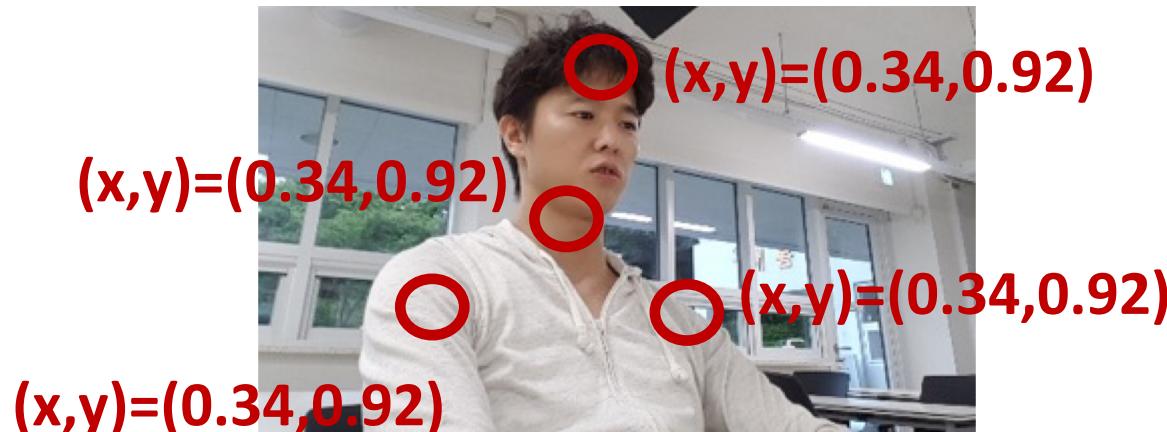
❖ **Pose estimation =**

Localization + Classification

Human Pose Estimation

❖ **Pose estimation =**

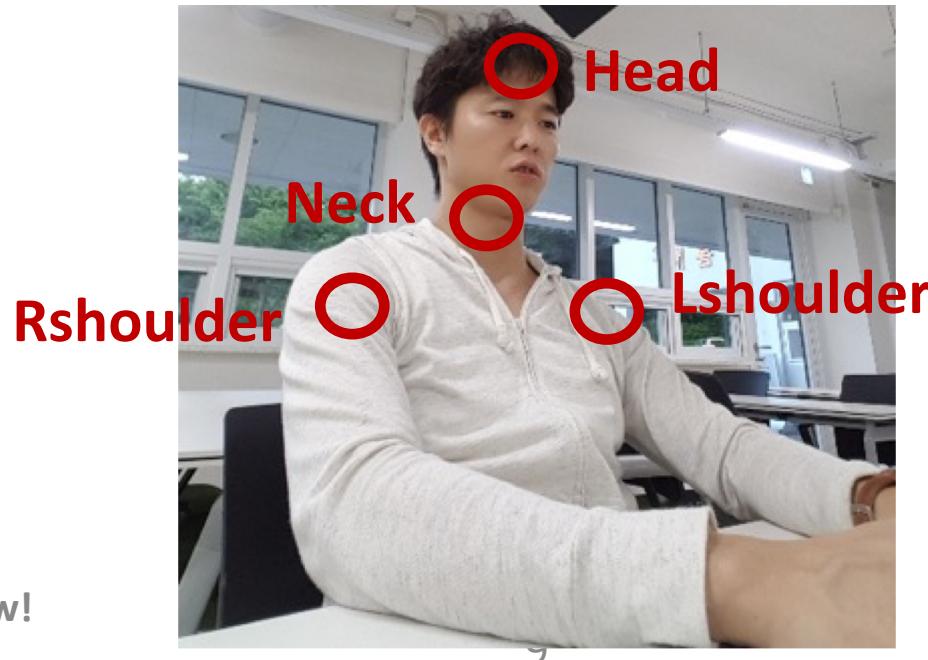
Localization + Classification



Human Pose Estimation

❖ **Pose estimation =**

Localization + Classification

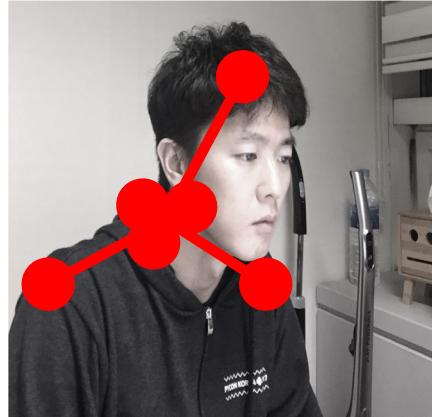


Human Pose Estimation



❖ Pose estimation

- **Localization**: 입력 image로부터 신체부위 (keypoint)의 위치를 찾는 일
- **Classification**: 찾은 신체 부위를 종류를 구분하는 일



Pose coordinate Prediction

- head=(0.1, 0.3)
- neck=(0.2, 0.6)
- RShoulder=(0.3, 0.1)
- LShoulder=(0.1, 0.9)



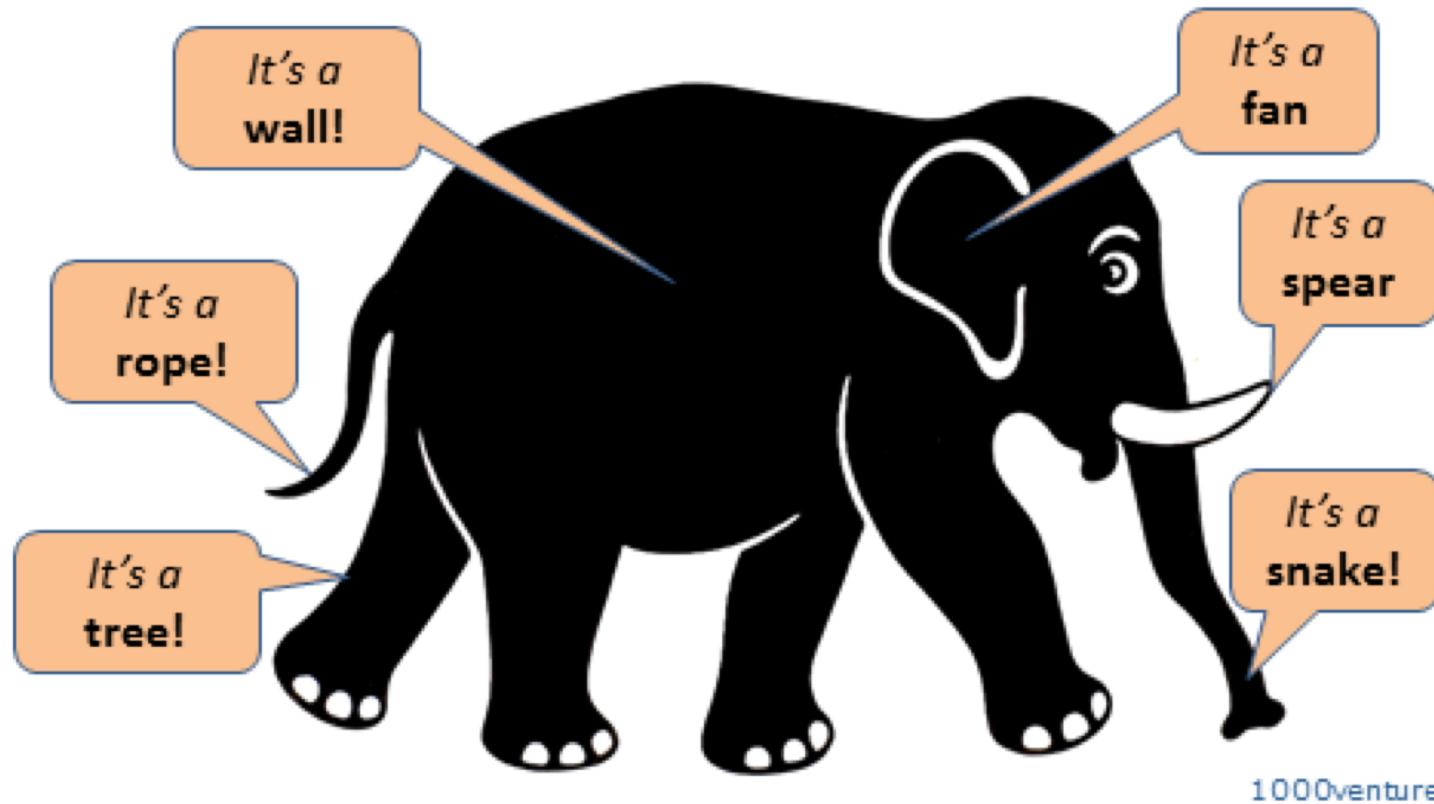
DeepPose (Alexander'14)

- ❖ 기존 방법: **Localization** → **classification**
 - **Localization**는 잘 하는거 같은데...

DeepPose (Alexander'14)



- ❖ 장님이 코끼리 만지는 것 처럼...



DeepPose (Alexander'14)



❖ 기존 방법: Localization → classification

- 신체의 일부가 가려져 있거나 보이지 않으면

....



DeepPose (Alexander'14)



- ❖ 기존 방법: **Localization** → **classification**
 - 신체의 일부가 가려져 있거나 보이지 않으면...
 - **Classification**을 잘 못함ㅠ
 - 이게 팔인가~ 다리인가~



DeepPose (Alexander'14)



- ❖ 기존 방법: **Localization** → **classification**
 - 신체의 일부가 가려져 있거나 보이지 않으면...
 - **Classification**을 잘 못함ㅠ
 - 이게 팔인가~ 다리인가~
 - 사람의 신체구조에 대한 이해 부족



DeepPose (Alexander'14)



- ❖ 핵심: **Holistic reasoning** + Coordinate Regression
 - 딥러닝을 이용하여 나무와 숲을 동시에 보자!
 - **Localization + classification**을 동시에 수행

Toshev, A., Szegedy, C., “Deeppose: Human pose estimation via deep neural networks,” CVPR 2014

DeepPose (Alexander'14)



- ❖ 핵심: **Holistic reasoning** + Coordinate Regression
 - 딥러닝을 이용하여 나무와 숲을 동시에 보자!
 - **Localization + classification**을 동시에 수행
 - Global 신체 구조를 이해하는 모델!
 - 머리는 목위에 있다!
 - 안보이는 body parts도 찾을 수 있게!
 - Classification 정확도 개선!

DeepPose (Alexander'14)

- ❖ 핵심: Holistic reasoning + **Coordinate Regression**
 - 목표: 입력 image X 를 2D 좌표 벡터 γ 로 Regression(회귀) 하는 함수를 학습하는 것

$$Y = N^{-1}(\psi(N(X); W))$$

DeepPose (Alexander'14)

❖ 핵심: Holistic reasoning + **Coordinate Regression**

- 목표: 입력 image X 를 2D 좌표 벡터 γ 로 Regression(회귀) 하는 함수를 학습하는 것

$$Y = N^{-1}(\psi(N(X); W))$$

$\psi(\cdot; W)$: 회귀 함수

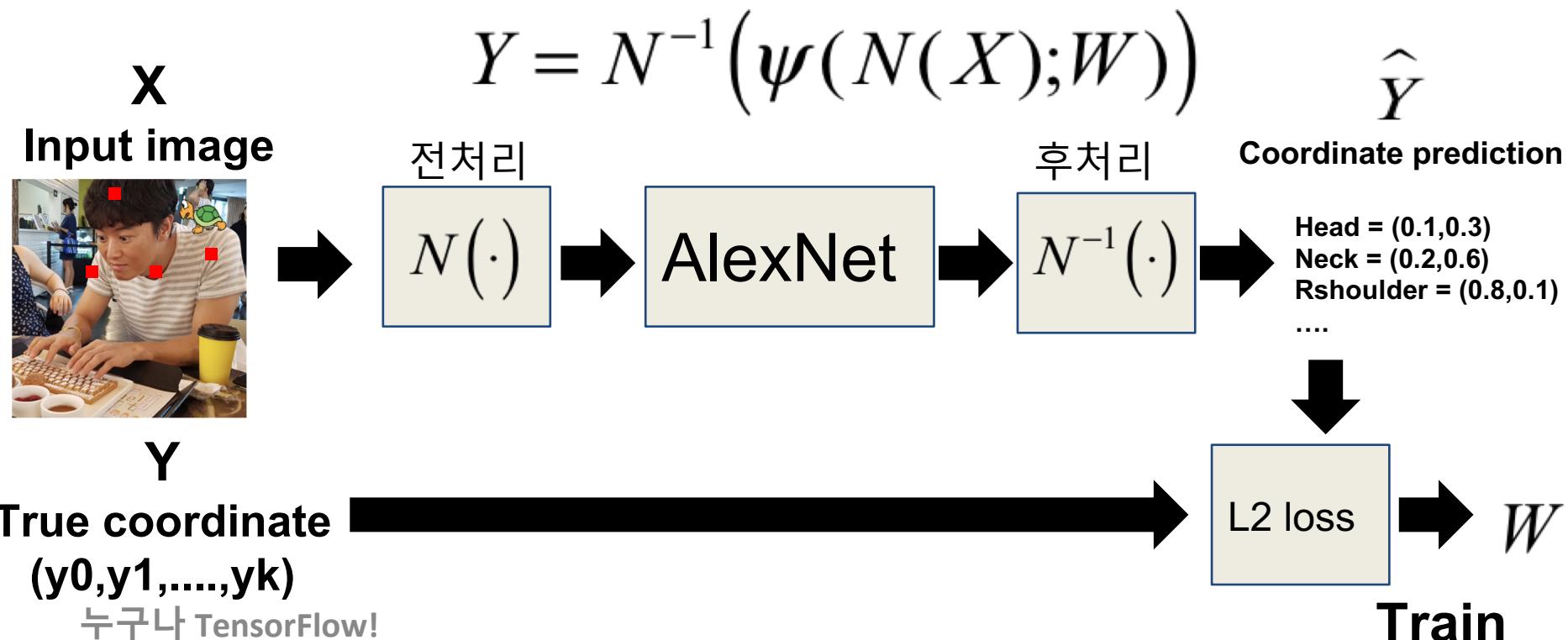
$N(\cdot)$: 전처리 함수

$N^{-1}(\cdot)$: 후처리 함수

DeepPose (Alexander'14)

❖ 핵심: Holistic reasoning + **Coordinate Regression**

- 목표: 입력 image X 를 2D 좌표 벡터 Y 로 Regression(회귀) 하는 함수를 학습하는 것



DeepPose (Alexander'14)

- ❖ Localization 정확성이 부족해ㅠ



Special thank to 피카소! :-)



DeepPose (Alexander'14)

- ❖ 개선: Pose displacement regression
 - Regressors의 직렬연결로 반복 개선

DeepPose (Alexander'14)

- ❖ 개선: Pose displacement regression



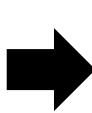
DeepPose (Alexander'14)



❖ 개선: Pose displacement regression

- Regressors의 직렬연결로 반복 개선
 - 각 body parts를 독립적으로 bounding box를 치고
 - 이전 prediction 와 True값의 차이(displacement)를 예측하여 개선

X with i-th box
from prev stage



전처리

$$N(\cdot)$$

Single stage regressor

$$\text{AlexNet}$$

후처리

$$N^{-1}(\cdot)$$

Displacement
prediction

$$\hat{\delta}$$



Displacement
from prev stage

$$\hat{\delta} = \hat{Y}_i^{S-1} - Y$$

누구나 TensorFlow!

J. Kang Ph.D.



$$\text{L2 loss}$$



$$W$$

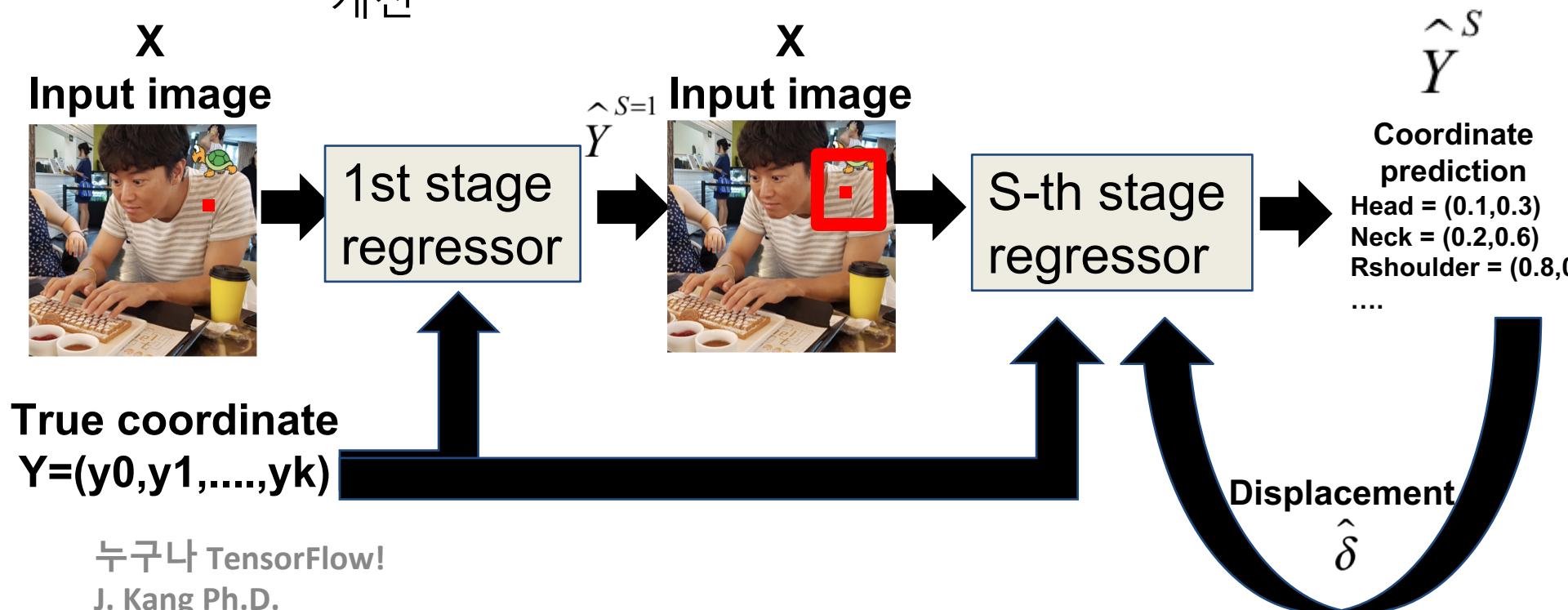
Train

DeepPose (Alexander'14)



❖ 개선: Pose displacement regression

- Regressors의 직렬연결로 반복 개선
 - 각 body parts를 독립적으로 bounding box를 치고
 - 이전 prediction 와 True값의 차이(displacement)를 예측하여 개선



Convolutional Heatmap Regressor

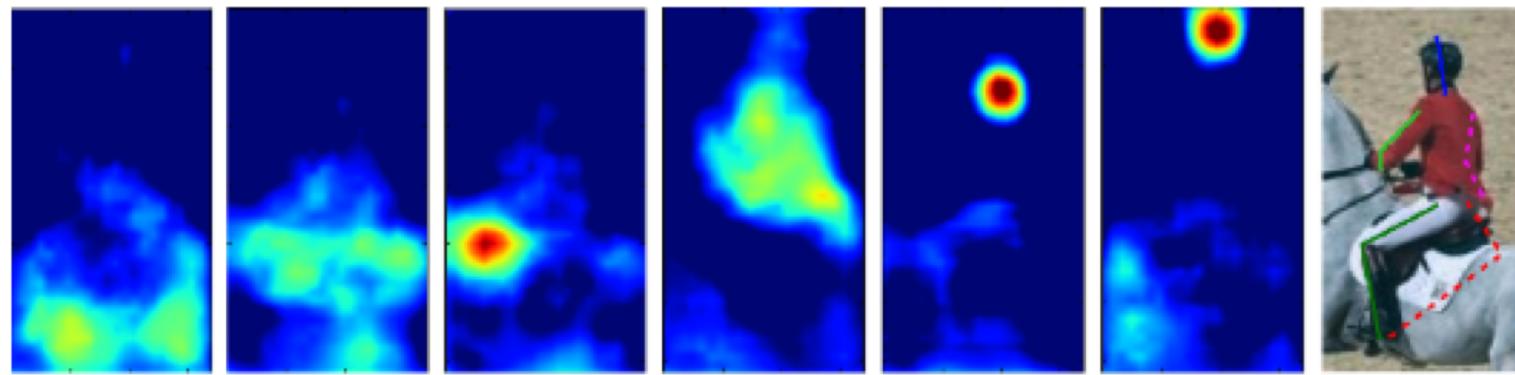
❖ Coordinate regression의 한계

- 좌표값만 출력하기 때문에 모델 prediction Y가 포함하는 표현력 (정보)가 적음
- 직렬연결 + 반복 coordinate prediction을 해도 충분한 location의 정확도 개선이 어려움
 - 이용할 수 있는 정보가 애당초 적으니깐!

Convolutional Heatmap Regressor

❖ 핵심: Use heatmap regression

- **Heatmap**: 2D Pixelwise likelihood map for part locations
- heatmap의 픽셀단위로 part localization의 confidence값을 제공
 - part localization의 full 정보를 제공하는 것임

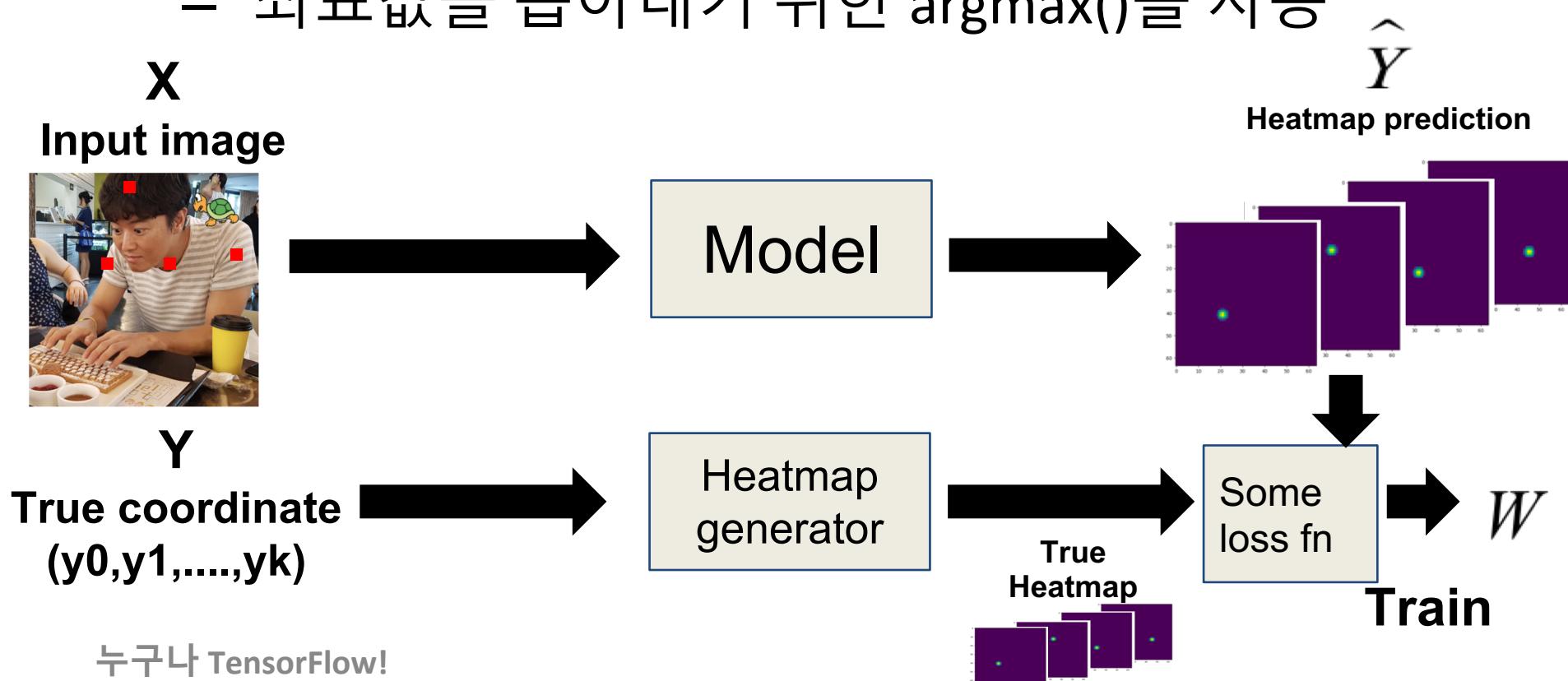


occluded ankle	occluded knee	visible knee	occluded wrist	neck	head	result
-------------------	------------------	-----------------	-------------------	------	------	--------

Convolutional Heatmap Regressor

❖ 핵심: Use heatmap regression

- 모델의 출력 값으로 heatmap을 사용
- 좌표값을 뽑아내기 위한 argmax()를 사용



Convolutional Heatmap Regressor

❖ 핵심: Use heatmap regression!!

- 다양한 형태의 heatmap regressor가 제안됨
- Not limited to the below:
 - [Efficient object localization using convolutional networks, CVRP 2015](#)
 - [Human pose estimation via convolutional part heatmap regression, ECCV 2016](#)
 - [Convolutional pose machines, CVRP 2016](#)
 - [Stacked hourglass network for human pose estimation, ECCV 2016](#)

Multiscale Understanding



❖ 문제: 다양한 크기의 사람 객체를 처리해야



[image credit: FLIC dataset](#)

Multiscale Understanding



- ❖ 문제: 다양한 크기의 사람 객체를 처리해야!
 - 다양한 크기의 receptive field를 가지도록 해야!

[image credit: FLIC dataset](#)

Multiscale Understanding



❖ 문제: 다양한 크기의 사람 객체를 처리해야



[image credit: FLIC dataset](#)

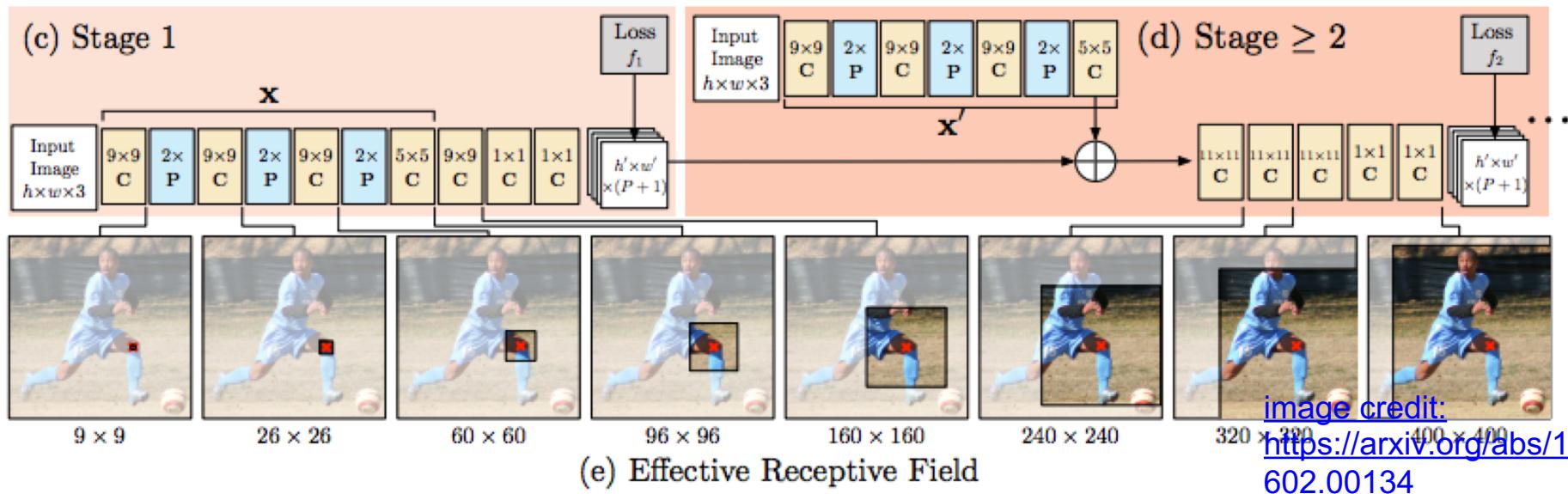
Multiscale Understanding



- ❖ 문제: 다양한 크기의 사람 객체를 처리해야!

- ❖ 해결방향1: Convolutional Pose Machine (Wei CVPR16)

- Multi-stage feature learning을 하여 effective receptive field를 점차 키워나간다.
- conv filter size를 다양하게 사용

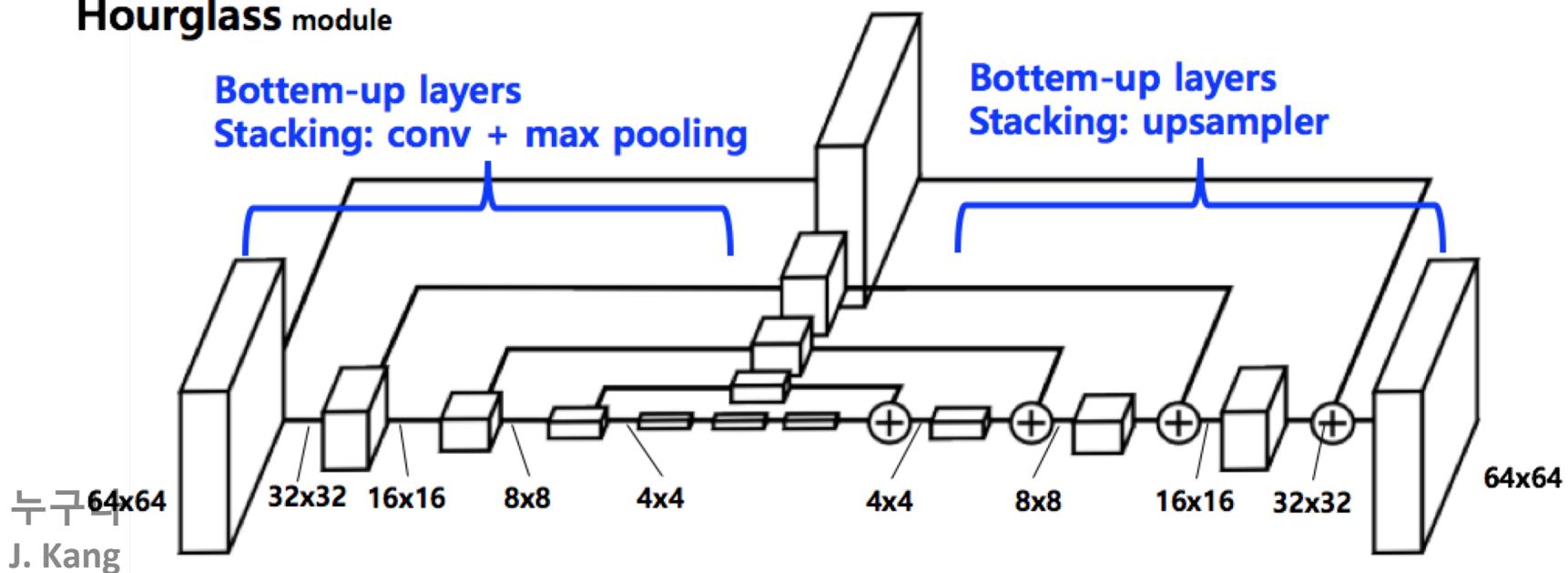


Multiscale Understanding



- ❖ 문제: 다양한 크기의 사람 객체를 처리해야 !
- ❖ 해결방향2: Stacked **Hourglass** (Newell, ECCV 16)
 - Multi-stage encoder-and-decoder 구조
 - Single channel pipeline

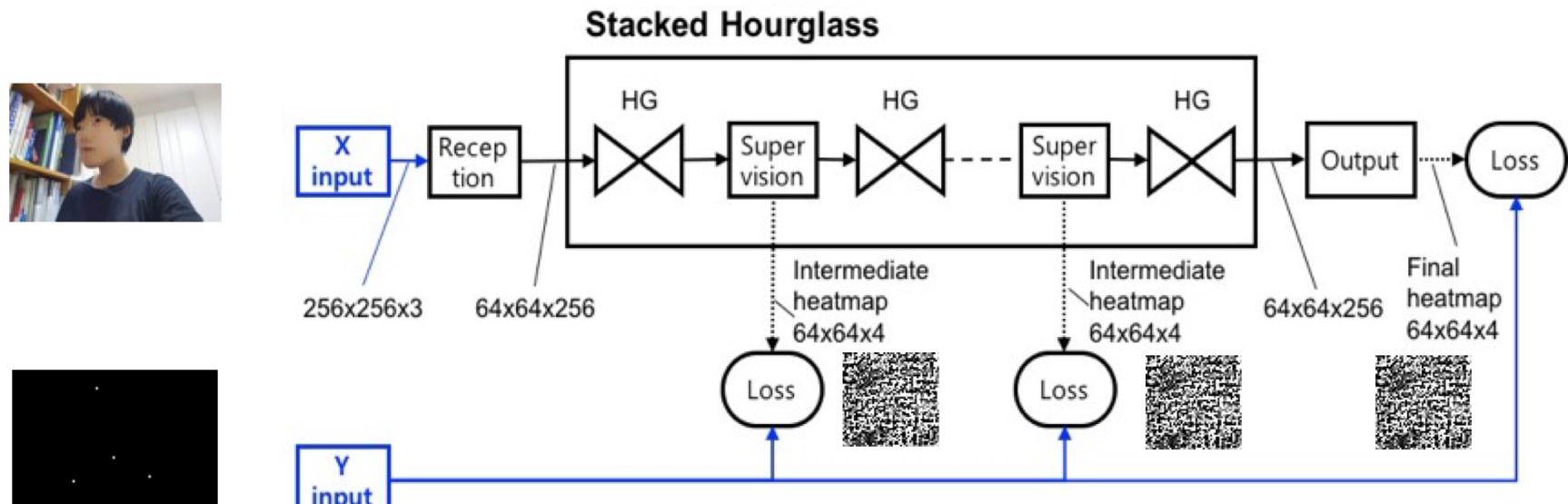
Hourglass module



Multiscale Understanding



- ❖ 문제: 다양한 크기의 사람 객체를 처리해야!
- ❖ 해결방향2: **Stacked** Hourglass (Newell, ECCV 16)
 - 신체 구조 이해를 위한 Hourglass stacking!
 - Iter



Beyond

❖ 이후 연구 발전

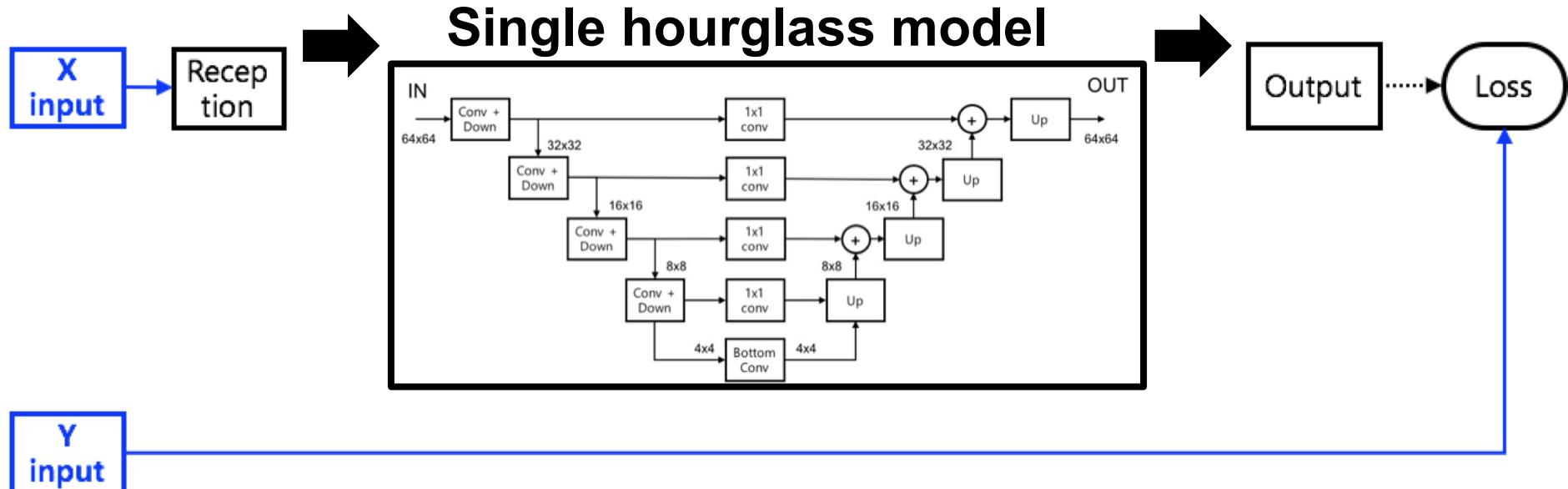
– TBU

Proj: Tiny Pose Est in Tensorflow


TensorFlow

- ❖ 프로젝트: 간단한 pose estimation 모델 구현하기

- <https://github.com/jwkanggist/tf-tiny-pose-estimation>



X input: pose image data

Y input: Ground truth heatmap

C: The number of channels

누구나 TensorFlow!

J. Kang Ph.D.

Proj: Tiny Pose Est in Tensorflow



- ❖ 프로젝트: 간단한 pose estimation 모델 구현하기
 - Single hourglass module을 구현해보자!
 - dataloader.py
 - eval.py
 - model_builder.py
 - trainer.py
 - model_config.py
 - train_config.py

Proj: Tiny Pose Est in Tensorflow

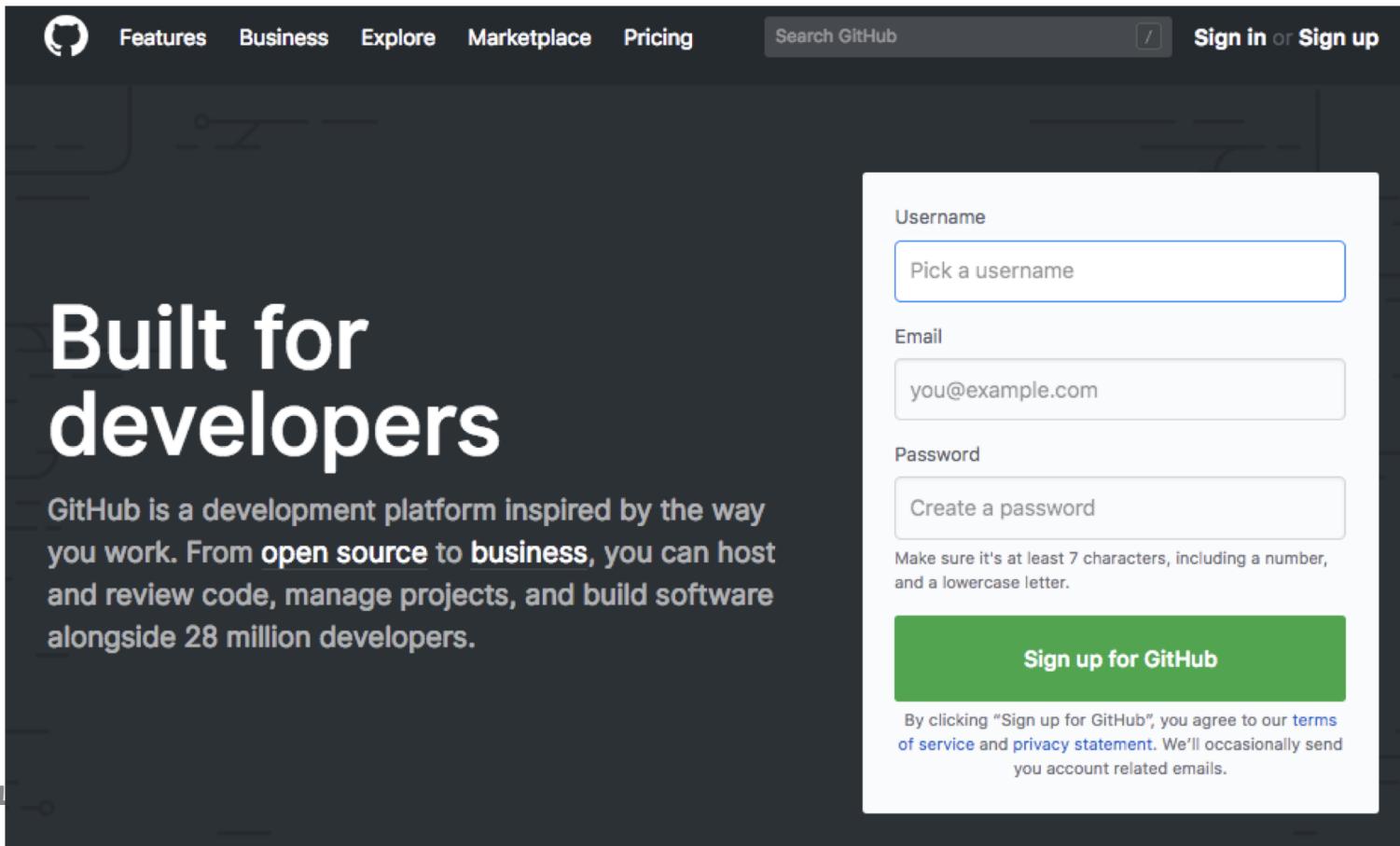


- ❖ 프로젝트: 간단한 pose estimation 모델 구현하기
 - Single hourglass module을 구현해보자!
 - `dataloader.py` (제공)
 - `eval.py (baseline 코드제공)`
 - `model_builder.py` (부분 제공)
 - `trainer.py (baseline 코드제공)`
 - `model_config.py (baseline 코드제공)`
 - `train_config.py (baseline 코드제공)`

Proj: Tiny Pose Est in Tensorflow



- ❖ 프로젝트: 간단한 pose estimation 모델 구현하기
 - git 계정 생성: www.github.com

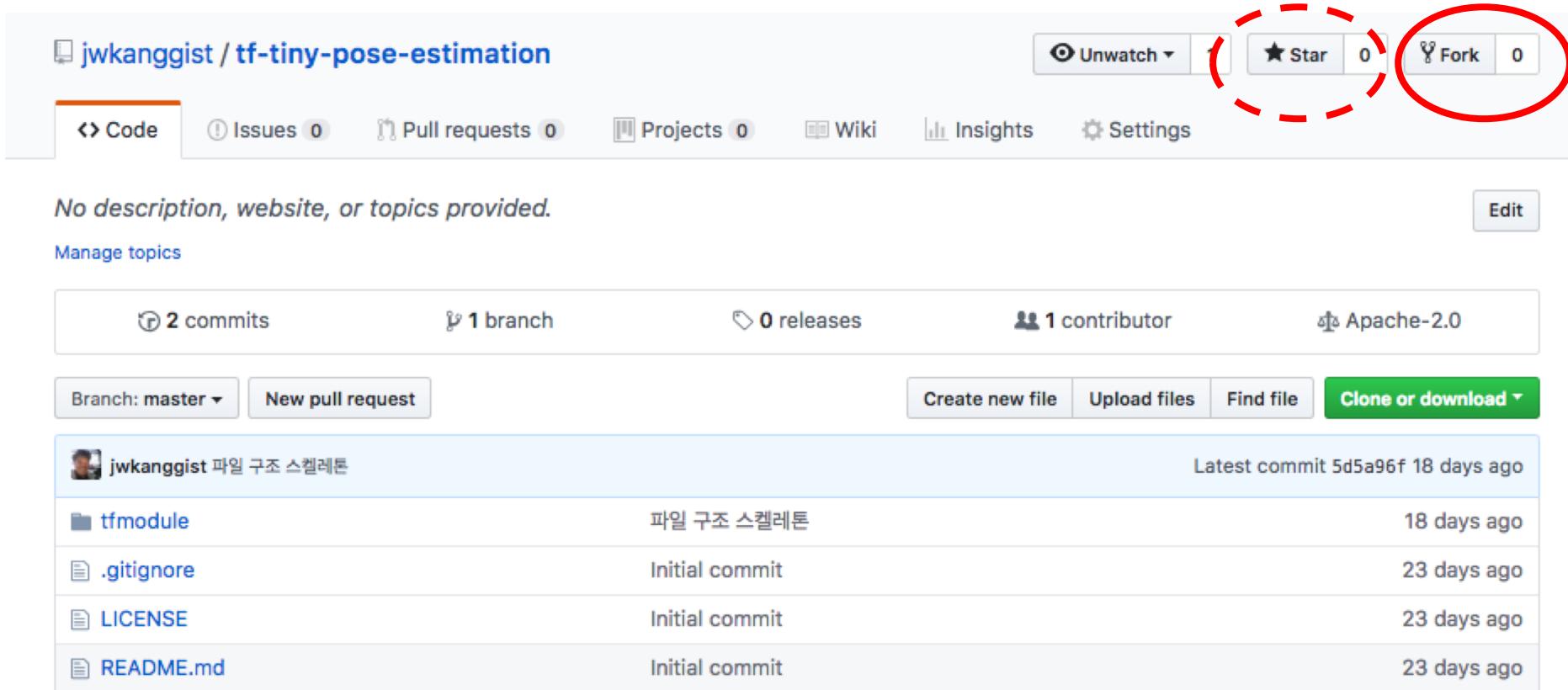


A screenshot of the GitHub sign-up page. The page has a dark background with white text. On the left, there is a large white text area that says "Built for developers" and "GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 28 million developers." At the top, there is a navigation bar with links for Features, Business, Explore, Marketplace, and Pricing. There is also a search bar and a "Sign in or Sign up" button. On the right, there is a sign-up form with fields for Username, Email, and Password. Below the form, there is a note: "Make sure it's at least 7 characters, including a number, and a lowercase letter." At the bottom of the form is a green "Sign up for GitHub" button. Below the button, there is a small note: "By clicking 'Sign up for GitHub', you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails."

Proj: Tiny Pose Est in Tensorflow



- ❖ 프로젝트: 간단한 pose estimation 모델 구현하기
 - fork [tf-tiny-pose-estimation](#) repo



jwkanggist / [tf-tiny-pose-estimation](#)

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

No description, website, or topics provided. Edit

Manage topics

2 commits 1 branch 0 releases 1 contributor Apache-2.0

Branch: master New pull request Create new file Upload files Find file Clone or download

File	Description	Committed
tfmodule	파일 구조 스켈레톤	18 days ago
.gitignore	Initial commit	23 days ago
LICENSE	Initial commit	23 days ago
README.md	Initial commit	23 days ago

Proj: Tiny Pose Est in Tensorflow



- ❖ 프로젝트: 간단한 pose estimation 모델 구현하기
 - git clone your repo

git downloadL: <http://git-scm.com/downloads>

[git fork는 웹에서]

```
$ git init
```

```
$ git branch -r
```

```
    origin/HEAD -> origin/master
```

```
    origin/develop
```

```
    origin/features/fb_proj_baselines
```

```
    origin/master
```

```
$ git checkout -t origin/features/fb_proj_baselines
```

Proj: Tiny Pose Est in Tensorflow



- ❖ 프로젝트: 간단한 pose estimation 모델 구현하기
 - Single hourglass module을 구현해보자!
 - 배운것을 써먹자
 - <https://goo.gl/5m93cg>

Proj: Tiny Pose Est in Tensorflow



- ❖ model_builder.py / model_config.py
 - model_config.py에서 파라미터값을 로드해서 입출력 shape를 결정
 - tf.slim API를 이용한 코딩

Proj: Tiny Pose Est in Tensorflow



- ❖ model_builder.py / model_config.py
 - Depthwise separable conv module:

Layer	Type	Kernel size	Stride	Activation
D4	batchnorm	-	-	-
D3	Conv	1x1	1	relu
D2	batchnorm	-	-	-
D1	Conv	3x3	1	None
In	Input	-	-	-

Proj: Tiny Pose Est in Tensorflow



- ❖ model_builder.py / model_config.py

- Downsample module:

Layer	Type	OutputMaps	Output Size	Filter size	Stride	Activation
DS2	max pool	32	N/2 x N/2	3x3	2	None
DS1	Depthwise Separable Conv	32	NxN	-	-	-
In	Input	32	NxN	-	-	-

Proj: Tiny Pose Est in Tensorflow



- ❖ model_builder.py / model_config.py
 - Upsample module:

Layer	Type	OutputMaps	Output Size
US2	Depthwise separable conv	32	NxN
US1	resize_bilinear	32	NxN
In	Input	32	N/2 x N/2

Proj: Tiny Pose Est in Tensorflow



- ❖ model_builder.py / model_config.py

- Reception module:

Layer	Type	OutputMaps	Output Size	Filter size	Stride	Activation
R4	maxpool	32	64x64	3x3	2	-
R3	Depthwise separable conv	32	128x128	-	-	-
R2	batchnorm	32	128x128	-	-	-
R1	Conv	32	128x128	7x7	2	relu
In	Input	3	256x256	-	-	-

Proj: Tiny Pose Est in Tensorflow



- ❖ model_builder.py / model_config.py

- Hourglass module:

Layer	Type	OutputMaps	Output Size	filter size	Activation
HU10	upsample	32	64x64		
HC9	conv → add skip connection from HD1 out	32	32x32	1x1	relu
HU8	upsample	32	32x32		
HC7	conv → add skip connection from HD2 out	32	16x16	1x1	relu
HU6	upsample	32	16x16		
HC5	conv → add skip connection from HD3 out	32	8x8	1x1	relu
HB4	Depthwise separable conv	32	8x8		
HD3	downsample	32	8x8		
HD2	downsample	32	16x16		
HD1	downsample	32	32x32		
In	Input	32	64x64		

Proj: Tiny Pose Est in Tensorflow



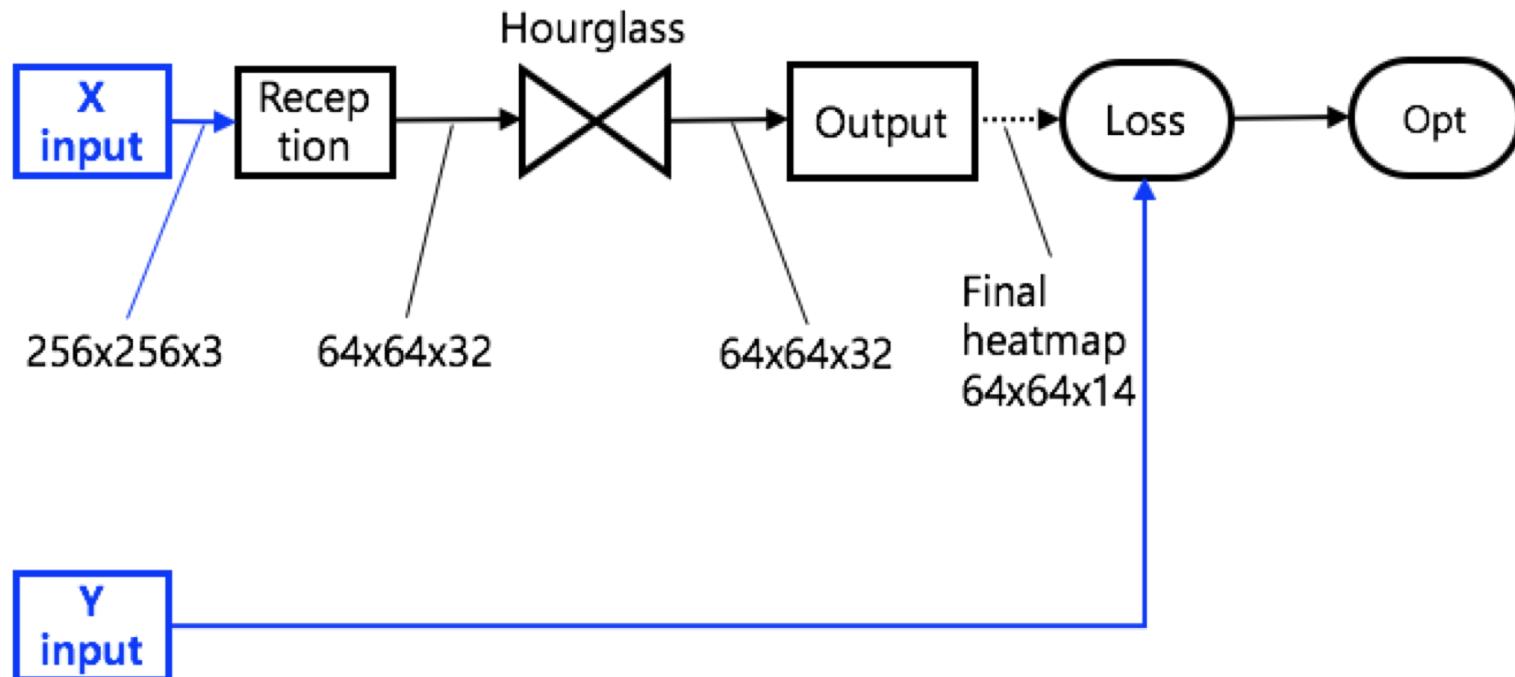
- ❖ model_builder.py / model_config.py
 - output module:

Layer	Type	OutputMaps	Output Size	Filter size	Stride	Activation
out	conv + dropout	14	64x64	1x1	1	None
In	Input	32	64x64	-	-	-

Proj: Tiny Pose Est in Tensorflow



❖ 전체 모델 구성도



X input: pose image data

Y input: Ground truth heatmap

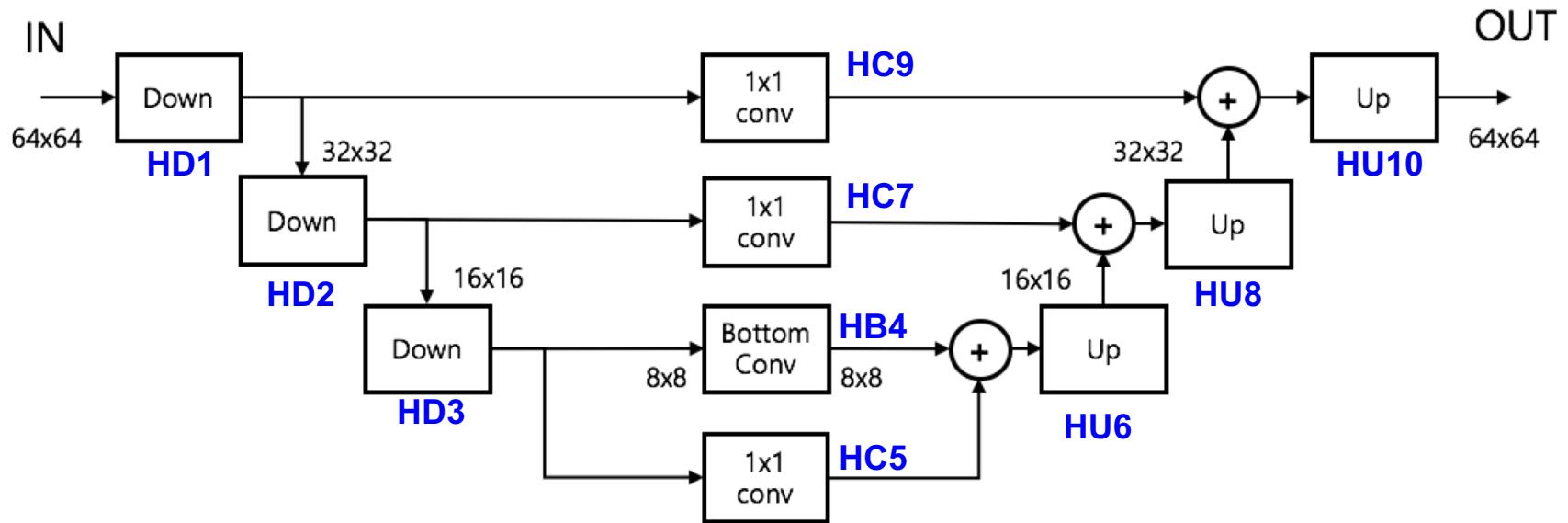
C: The number of channels

Proj: Tiny Pose Est in Tensorflow



Single Hourglass Module for tf-tiny-pose-estimation

- Stage Num: 3



Tiny Pose Est in Tensorflow



- ❖ `trainer.py / train_config.py`
 - `tf.device` 를 이용한 GPU사용 설정
 - `tf.data.Iterator`를 이용한 data feeding
 - `train_config.py`를 통한 훈련파라미터 설정
 - tensorboard 사용설정

Proj: Tiny Pose Est in Tensorflow



- ❖ `trainer.py / train_config.py`
 - `tf.device` 를 이용한 GPU사용 설정
 - training set만 사용해서 훈련만 수행
 - `tf.data.Iterator`를 이용한 data feeding
 - `train_config.py`를 통한 훈련파라미터 설정
 - ckpt 저장
 - tensorboard 사용설정

Proj: Tiny Pose Est in Tensorflow



❖ eval.py

- batch_size = 1
- validation set 사용
- ckpt로 부터 그래프 로드
- loss 함수 optimizer 함수 제거
- 평가 메트릭으로 성능 측정

Awesome Pose estimation Repo



- [Open pose](#) (ECCV2016 /CVPR2017, CMU)
- [Multi-person pose estimation](#) (CVPR2017)
- [tf-pose-estimation](#) (카카오 일두님)

- [paper 정리 repo](#) ([cbsudux](#) 's repo)

AWS 실행방법

```
$ git clone <your git http>
$ git init
$ git branch -r
$ git checkout -t origin/features/fb_proj_baselines
$ cd tf-tiny-pose-estimation
$ cd tfmodules
$ python trainer.py
$ tensorboard --logdir ./export
```

Local

```
$ git commit -m "first comment"
$ git add .
$ git push
```

Note

1) master branch 버그 픽스 (필수)

- git checkout master
- git reset --hard HEAD
- git pull

1) ai_challenge.zip full dataset download (필수)

- a) ./dataset에서 replace
- b) train_config에서 self.train_data_size로 조절

2) pycocotool WIN 버전 깔기

3) tensorboard --logdir ./export

4) 각자 하고싶은 수준 정하기

모두연 MoT랩 소개



- ❖ 딥러닝을 활용하여 세상을 이롭게 할 IoT/Mobile App 개발에 대한 연구를 같이 해봐요!!
- ❖ <https://www.facebook.com/lab4all/posts/761099760749661>
- ❖ jwkang10@gmail.com로 메일

- ❖ Keywords:
 - Thin CNN Model
 - Model Pruning
 - Tensorflow + lite
 - Embedded Sys. (IoT)
 - Android Mobile/Things

