# CLIMAX - RADAR

Bishal

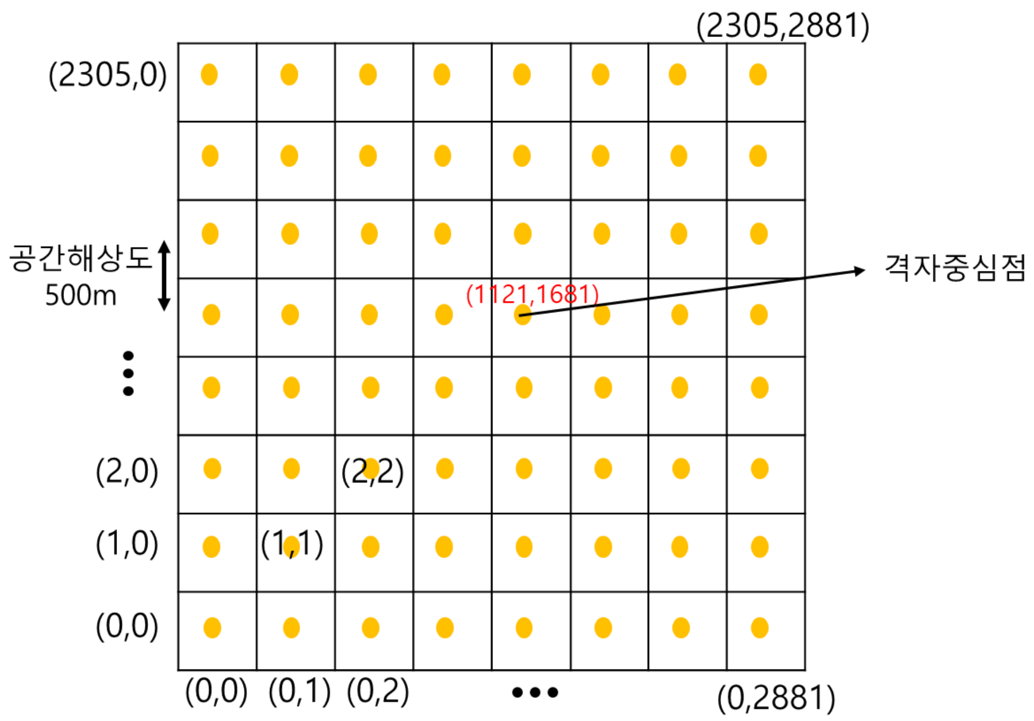
Computer Vision & Pattern Recognition Lab

kit 금오공과대학교
Kumoh National Institute of Technology

# DATA INFORMATION- 레이더 합성자료

- 수집한 레이더 합성자료는 10분 간격의 500m 공간해상도 자료임
- 레이더 합성자료는 값정보가 좌하단에서 우상단순으로 기록되어있음
- 레이더 합성자료의 격자는 (y,x) -> (2305,2881)으로 구성되어있음
- 기준 격자점은 (1121,1681)
- 기준 위경도는 N 38.0 , E 126.0
- 투영방법 : LCC (Lambert conformal conic projection)
- 원본 .bin 파일의 값정보

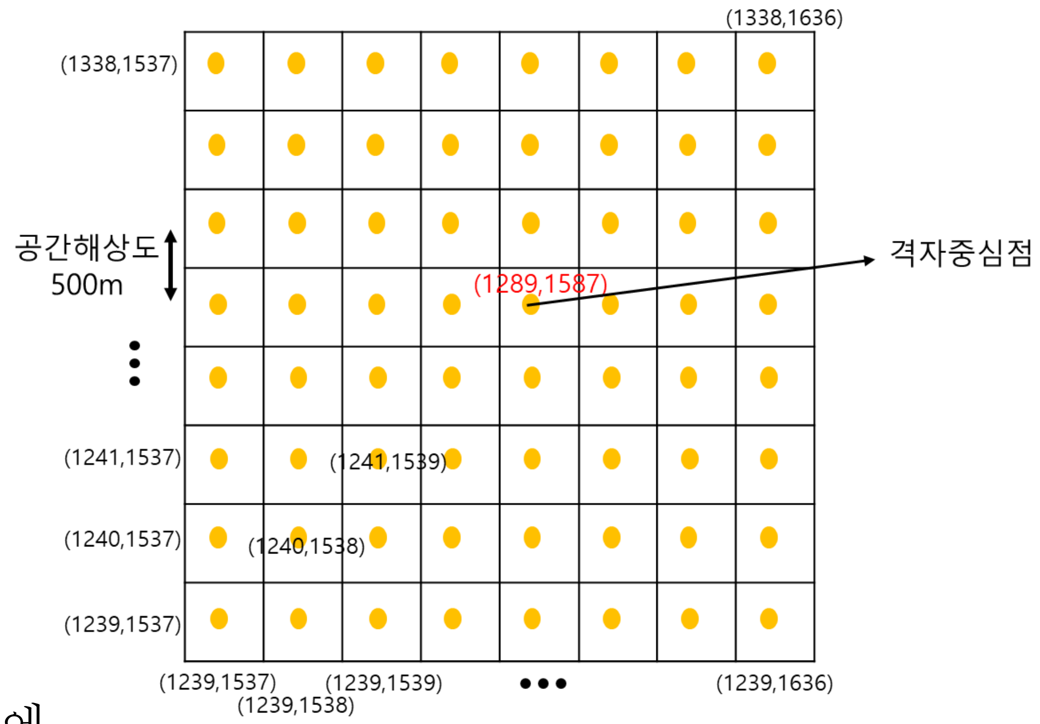
  - 관측영역내 표시를 위한 최소값 : -20000
  - 관측영역내 비관측영역 NULL 값 : -25000
  - 관측반경 밖 NULL값 : -30000
  - 값 X 100 : 강우강도



# PRE-PROCESSING - 레이더 합성자료(HSP) 전처리

- (2305, 2881) 격자의 원본 자료를 (100,100) 크기의 격자로 축소
- 서울시 범위의 학습을 위하여 원본 데이터의 (1289, 1587) 격자를 기준으로구축
- 강우강도 값 처리

| 원시값(int16) | 의미 | 처리 결과 |
|---|---|---|
| > -20000 | 유효한 강우값 | 값/100.0 ->float32 |
| -20000 이하 | 특수값(결측 등) | np.nan |
| -20000 | 관측값 중 최소값<br>(약하거나 없음) | NaN |
| -25000 | 관측 불가 | NaN |
| -30000 | 레이더 반경 외 | NaN |




- 단위를 줄이기위하여 원본 값은 100이 곱해진 상태로 저장되어있음 (그래서 100으로 나눠야함)
- 기상청에서 제공하는 위경도파일(netCDF)를 활용하여 격자에 해당하는 위경도 표시
- 좌표계 : LCC (평면) ,WGS84 (지리좌표계)

# PRE-PROCESSING - 레이더 합성자료(HSP) 전처리

- HSP 레이더 자료와 전처리 과정은 동일하지만 값처리 방식이 다름
- HSR은 반사도(dBZ) 자료로 강우강도(mm/hr)로의 변환이 필요
- HSP와 동일하게 저장된 값은 100이 곱해진 값
- 반사도 값 처리

| 원시값(int16) | 의미 | 처리 결과 |
|---|---|---|
| -25000 | 비관측 영역 | np.nan |
| -30000 | 관측 반경 외 | np.nan |
| 그 외 | 유효한 dBZ값 | 값/100.0 ->float32 |

처리된 반사도(dBZ) 값을 선형 반사도(Z) 로 변환 $\Rightarrow Z = 10^{\frac{dBZ}{10}}$

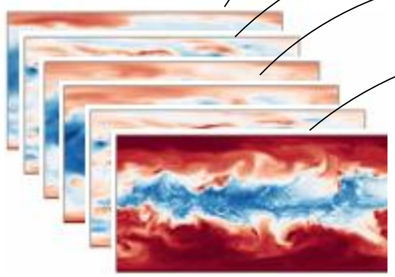
Z-R 관계식을 통해 강우강도 추정(Marshall-Palmer 공식) $Z = a \cdot R^b \ (a = 200, b = 1.6)$

- 선형반사도 Z의 단위는 mm^6/m^3

강우강도(mm/hr) 변환 후 이전의 HSP 전처리 방식과 동일하게 100X100크기로 축소하여 csv로 저장

# CLIMAX MODEL PREPARATION

ClimaX is a grid-to-grid model. The model expects its input to be a set of 2D grids.



Each grid represents a different physical variable

① These variables are treated like the "channels" of a multi-channel image

② The model learns spatial relationships between data points by not learning x, y coordinates but by its embedded positional embedding. The model breaks grid into patches, and it learns a unique embedding for each patch's position

# DATA PREPARATION

The saved .csv files in Slide 4, contain coordinate information (x, y, lat, lon) and a rain value.

--- First 5 Rows (Head) ---

| | x | y | rain | lat | lon |
|---|---|---|---|---|---|
| 0 | 1239 | 1537 | NaN | 37.336834 | 126.688200 |
| 1 | 1240 | 1537 | NaN | 37.336796 | 126.693985 |
| 2 | 1241 | 1537 | NaN | 37.336754 | 126.699770 |
| 3 | 1242 | 1537 | NaN | 37.336710 | 126.705550 |
| 4 | 1243 | 1537 | NaN | 37.336674 | 126.711334 |

hsp_202306221220.csv

Based on ClimaX requirements as detailed in previous slide, we use x, y, lat, and lon columns to build the 2D grid and rain as the feature channel.

We are provided with three types of rain features – HSP, HSR and Reflectivity (HSR_dBZ). So, we use these rain features to make grids for model training.

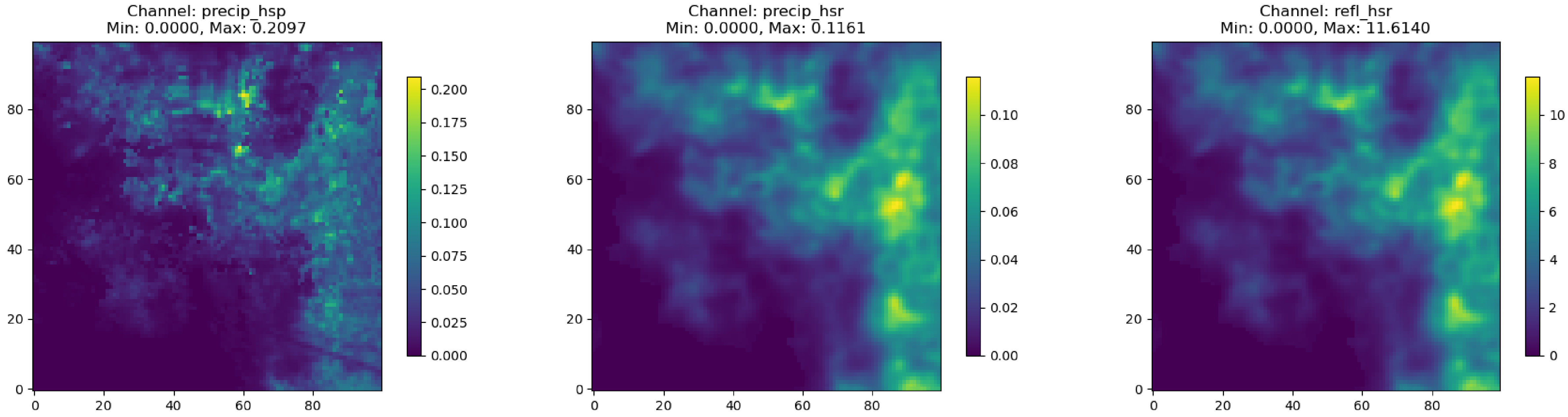
## Missing Values

| Date | Hsp | Hsr | Refl |
|---|---|---|---|
| 20210604 | 143 | 144 | 144 |
| 20210701 | 143 | 144 | 144 |
| 20210702 | 143 | 144 | 144 |
| 20211103 | 144 | 126 | 126 |
| 20211130 | 144 | 139 | 139 |
| 20211213 | 118 | 144 | 144 |
| 20211214 | 117 | 144 | 144 |
| … | … | … | … |

→ Skipped days with that had missing information.

# DATA PREPARATION

We used the *pandas.pivot_table* function to correctly transform the coordinate-list data into the 100x100 2D grids with rain data variable as required by the ClimaX model.



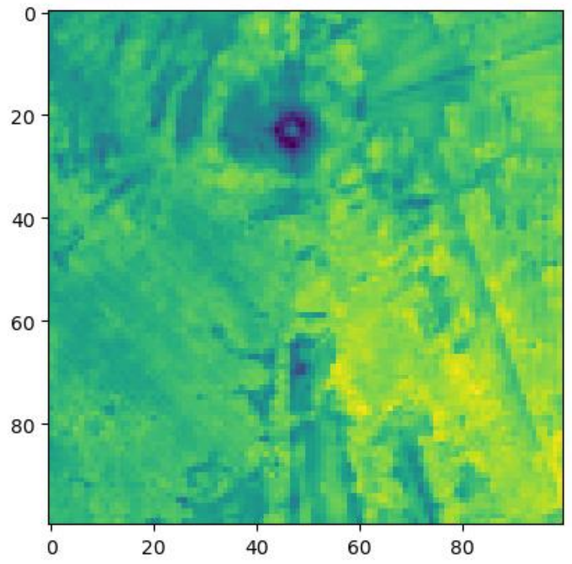

202409202350_merged.nc

Normalization means {'precip_hsp': 0.00189739, 'precip_hsr': 0.00140531, 'refl_hsr': 0.14053132}
Normalization stds: {'precip_hsp': 0.00323984, 'precip_hsr': 0.00205202, 'refl_hsr': 0.20520198}

# DATA PREPARATION

## Climatology Map

It is made by looping through all the files in the train directory, sums them up, and then divides by the total number of samples to get the average map.




Average visual map of all the training data in the dataset.

(Required by ClimaX model)

# DATA PREPARATION

Conversion and pre-processing of data to ClimaX model format

```
preprocess/
├── hsp/
│    ├── hsp_202106010000.csv
│    ├── hsp_202106010010.csv
│    ├── hsp_202106010020.csv
│    ├── …
├── hsr(dBZ)/
│    ├── hsr_202106010000.csv
│    ├── hsr_202106010010.csv
│    ├── …
├── hsr(rainfall)/
│    ├── hsr_202106010000.csv
│    ├── …
```

```
data/
├── all_merged/
│    ├── image_logs/
│    │    ├── train/
│    │    │    ├── 202106010250.npz
│    │    │    ├── 202106010500.npz
│    │    │    ├── …
│    │    │    ├── climatology.npz
│    │    ├── test/
│    │    ├── val/
│    │    ├── lat.npy
│    │    ├── lon.npy
│    │    ├── normalize_mean.npz
│    │    ├── normalize_std.cpz
│    │    ├── split_info.json
```

properties

```
Dimensions:    (time: 144, y: 100, x: 100)
Coordinates:
  * time       (time) datetime64[ns] 1kB
  * y          (y) int64 800B
  * x          (x) int64 800B
    lat        (y, x) float64 80kB ...
    lon        (y, x) float64 80kB ...
Data variables:
    precip_hsp (time, y, x) float32 6MB ...
    precip_hsr (time, y, x) float32 6MB ...
    refl_hsr   (time, y, x) float32 6MB ...
```

금오공대

# CLIMAX MODEL TRAINING

Splitting data into training, validation and test sets.

```
data/
├── all_merged/
│    ├── image_logs/
│    │    ├── train/
│    │    │    ├── 202106010250.npz
│    │    │    ├── 202106010500.npz
│    │    │    ├── …
│    │    ├── test/
│    │    ├── val/
│    │    ├── lat.npy
│    │    ├── lon.npy
│    │    ├── normalize_mean.npz
│    │    ├── normalize_std.cpz
│    │    ├── split_info.json
```

Total number of data points ⇒ 188071

Training Set (70%)

2021-06-01 → 2023-12-04

Training data count ⇒ 131649

Validation Set (20%)

2024-08-22 → 2025-08-22

Validation data count ⇒ 37614

Test Set (10%)

2023-12-05 → 2025-03-01

Test data count ⇒ 18808

# CLIMAX MODEL TRAINING

Preprocessing configuration.

preprocessing.py

```
SOURCE_CONFIG = {
    'precip_hsp': {'path': '/mnt/ext/preprocess/hsp',          'prefix': 'hsp'},
    'precip_hsr': {'path': '/mnt/ext/preprocess/hsr_rainfall', 'prefix': 'hsr'},
    'refl_hsr'  : {'path': '/mnt/ext/preprocess/hsr_dBZ',       'prefix': 'hsr'}
}
```

```
preprocess/
├── hsp/
│   ├── hsp_202106010000.csv
│   ├── hsp_202106010010.csv
│   ├── hsp_202106010020.csv
│   ├── …
├── hsr(dBZ)/
│   ├── hsr_202106010000.csv
│   ├── hsr_202106010010.csv
│   ├── …
├── hsr(rainfall)/
│   ├── hsr_202106010000.csv
│   ├── …
```

# CLIMAX MODEL TRAINING

## Training Configuration (training from scratch)

Model:  climax.regional_forecast.arch.RegionalClimaX
Image size: [100, 100]
Patch size: 4

embeded dimension: 1024
Depth: 8
Num heads: 16

Drop rate: 0.1

Region: KoreaSeoul

climax/src/climax/utils/data_utils.py

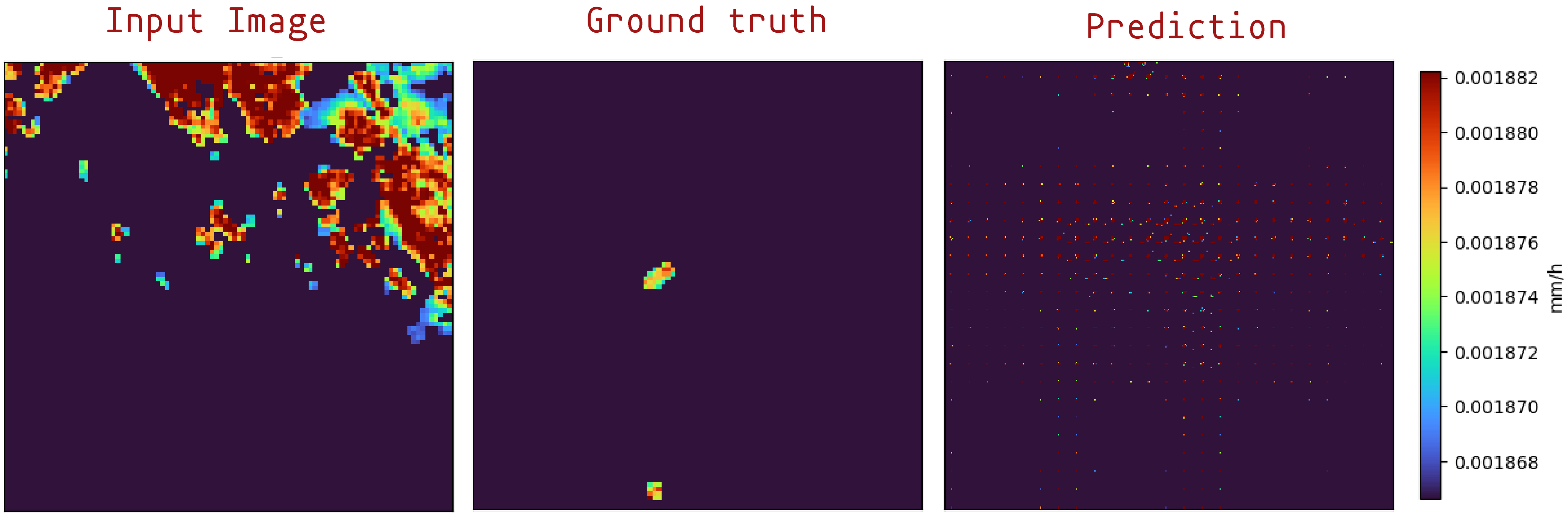
```
},
'Australia': { # 10x14
    'lat_range': (-50, 10),
    'lon_range': (100, 180)
},
'KoreaSeoul': {
    # This is a specific region for Korea, based on the lat/lon ranges of data
    "lat_range": (37.25, 37.90),
    "lon_range": (126.60, 127.35),

},
'Global': { # 32, 64
    'lat_range': (-90, 90),
    'lon_range': (0, 360)
}
```

Epochs: 10
Batch size: 32
Predict range: 72 # in hours

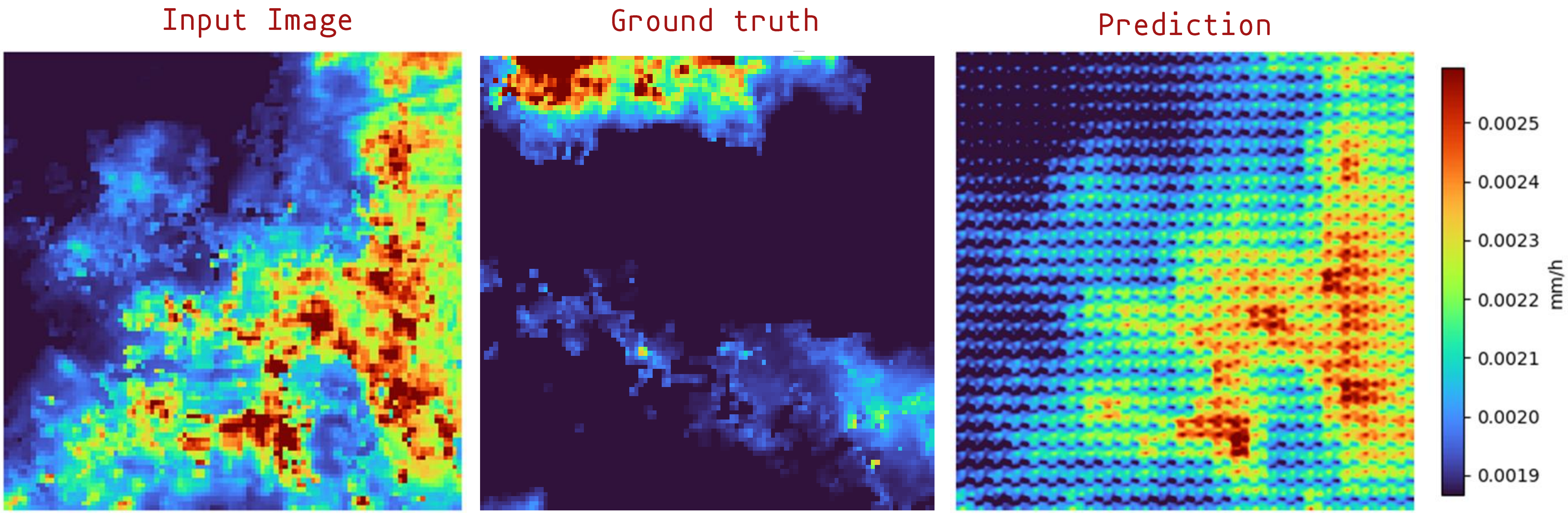
Input variables: precip_hsp
Output variables : precip_hsp

Test Accuracy → 56.69%
MSE → 2.74
Time/epoch → 40 mins (single GPU)

# SAMPLE PREDICTION OUTPUT MAPS



# SAMPLE PREDICTION OUTPUT MAPS



# CLIMAX MODEL TRAINING

Training Configuration (training from scratch)

Model: climax.regional_forecast.arch.RegionalClimaX
Image size: [100, 100]
Patch size: 4

embeded dimension: 1024
Depth: 8
Num heads: 16

Drop rate: 0.1

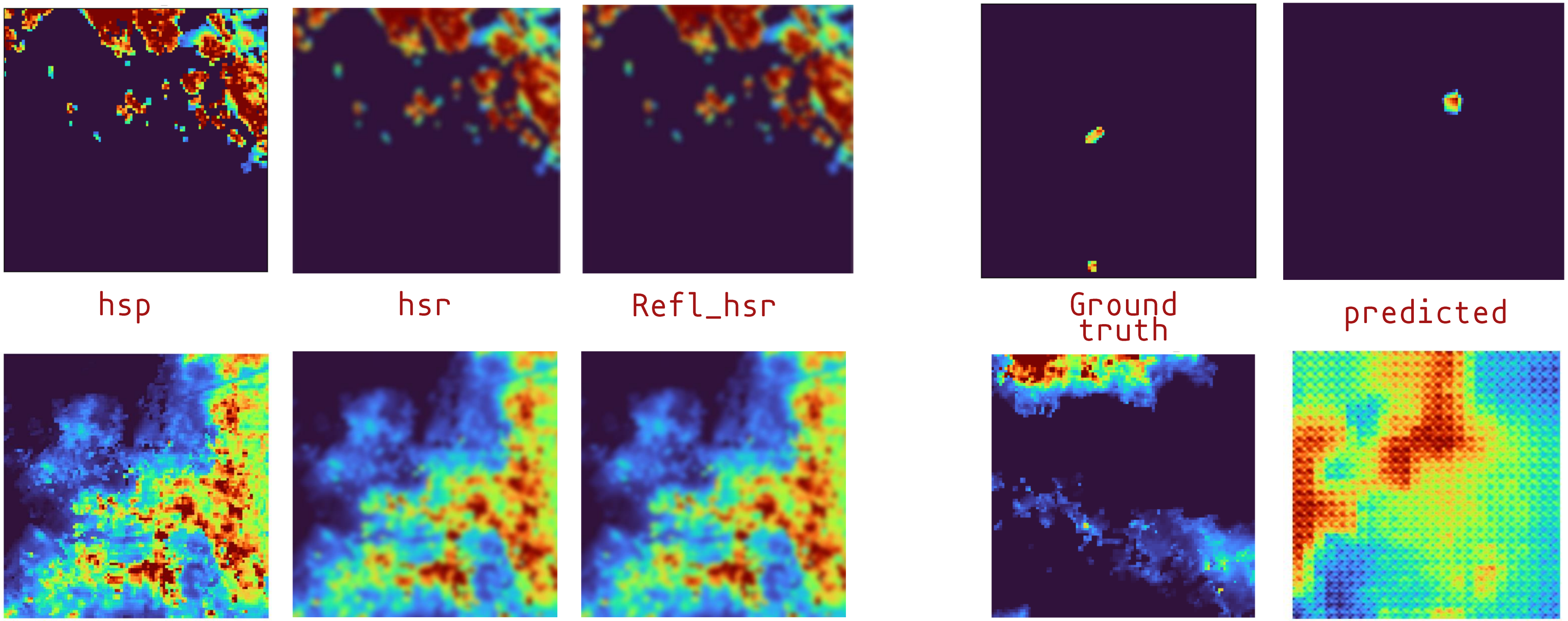
Region: KoreaSeoul

Epochs: 10
Batch size: 32
Predict range: 72 # in hours

Input variables: precip_hsp, precip_hsr, refl_hsr
Output variables : precip_hsp

Test Accuracy → 62.07%
MSE → 2.11
Time/epoch → 40 mins (single GPU)

# SAMPLE PREDICTION OUTPUT MAPS



## Planned Experiments

### Fixed Time Experiments

| Input Variables | Output Variables |
|---|---|
| precip_hsp | precip_hsp |
| precip_hsr | precip_hsr |
| refl_hsr | refl_hsr |
| precip_hsp, precip_hsr, refl_hsr | precip_hsp |
| precip_hsp, precip_hsr, refl_hsr | precip_hsr |
| precip_hsp, precip_hsr, refl_hsr | refl_hsr |
| precip_hsp, precip_hsr, refl_hsr | precip_hsp, precip_hsr, refl_hsr |

### Variable Time Experiments

Use the best model in above experiments to train the model to predict {24, 48, 72} hours

# Installation and Setup

GitHub Code development and version control (Share GitHub User ID for code updates)

## Installation

1. **Clone this repository (including ClimaX submodule)**

```
git clone  https://github.com/bluesaiyancodes/KlimaX.git
cd KlimaX
```

2. **Create and activate conda environment**

```
cd ClimaX
conda env create --file docker/environment.yml
conda activate climaX
```

3. **Install radar rainfall fork package**

```
pip install -e .
```

*(To be updated with additional dependencies as the project evolves)*

# CLIMAX MODEL TRAINING

## Usage

### 1. Preprocessing

```
python preprocessing.py --dataset_type all
```

### 2. Training

- For training the global model:

```
cd ClimaX
python src/climax/global_forecast/train.py --config configs/custom/global_radar_forecast_scratch.yaml
```

- For training the regionsal model (localized to Seoul):

```
python src/climax/regional_forecast/train.py --config configs/custom/regional_radar_forecast_scratch.yaml
```

### 3. Training Results

> The training logs are saved and readable through `TensorBoard`.

- View results on `TensorBoard`. Use `--bind_all` tag if the experiments are perfomed on server.

```
tensorboard --logdir outputs/radar_forecast_scratch/logs/version_34_train1/ --bind_all
```

- Get raw results from TensorBoard and save them to files. The raw results are stored in `.csv` file formats.

```
cd ../
python ex_tb.py --log_dir ClimaX/outputs/radar_forecast_scratch/logs/version_34_train1/ --gen_dir extracted_results/version_34_train1/
```

extracted_results / version_34_train1
- epoch.csv
- lr-AdamW_pg1.csv
- lr-AdamW_pg2.csv
- test_acc_precip_hsp_3_days.csv
- test_acc.csv
- test_w_mse_precip_hsp_3_days.csv
- test_w_mse.csv
- test_w_rmse_precip_hsp_3_days.csv
- test_w_rmse.csv
- train_loss.csv
- train_precip_hsp.csv
- val_acc_precip_hsp_3_days.csv
- val_acc.csv
- val_w_mse_precip_hsp_3_days.csv
- val_w_mse.csv
- val_w_rmse_precip_hsp_3_days.csv
- val_w_rmse.csv