

그래서 비가온다go?

강우 예측 정확도를 높이는 딥러닝 모델 개발

한국생산성본부 / 정보통신기획평가원
이미지 분석 기반의 인공지능 플랫폼 개발자 양성과정



팀명 : 오조오억번

팀원 : 이동석/최서연/정인서/신다연

한국생산성본부 / 정보통신기획평가원

이미지 분석 기반의 인공지능 플랫폼 개발자 양성과정

오조오역번 팀원 소개



이동석
Dong-suk Lee

발표자
프로젝트 기획
데이터 시각화
기존 모델 분석
향후 활용방안 아이디어



최서연
Seo Yeon Choi

프로젝트 기획
도메인 분석
데이터 수집 및 준비
모델 설계/튜닝/검증
최종 프로젝트 보고서 작성



정인서
In-Seo Jung

프로젝트 기획
개발 계획서 작성
데이터 수집 및 준비
기존 모델 분석
모델 파라미터 튜닝/검증



신다연
Da Yeon Shin

팀 프로젝트 리더
프로젝트 기획
데이터셋 전처리
향후 활용방안 아이디어

- 지구온난화와 더불어 집중호우, 태풍과 같은 위험기상이 빈발

→ **정확한 기상정보의 신속한 제공에 대한 수요 급증**

- 최근 동일 학습 자료량에 대한 딥러닝 성능이 큰 폭으로 향상

- 시간별 레이더 구름 반사도 이미지 빅데이터를 분석 및 활용

→ **높은 정확도의 강우 예측 딥러닝 모델을 확보**

- 향후 재난안전관리, 기상청, 한수원 등 다양한 공공분야에서 활용 가능

[빅데이터] 올 여름 '일기예보 오보' 검색
3배 급증, 국민 불만 극에 달해

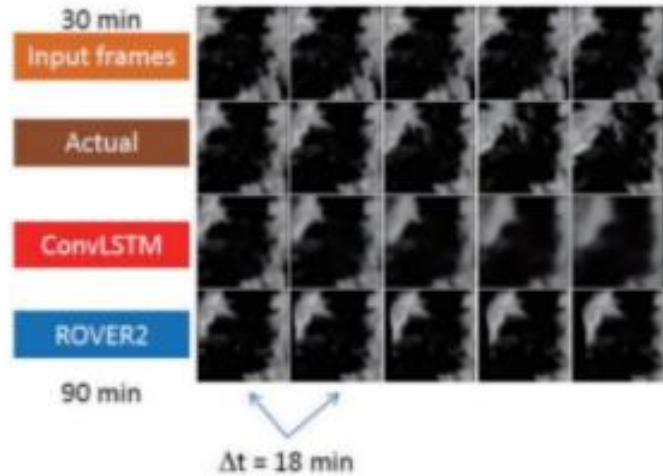
배포 2020-09-05 14:58:30 | 수정 2020-09-05 15:32:38 |



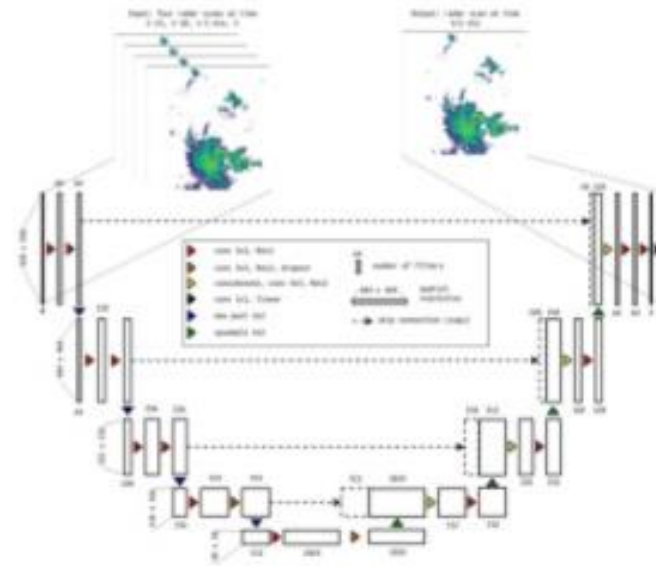
'역대급 폭염 예보...비만 왔다' 기상청 오보에
불만 쌓여

지구온난화-이상기후로 진땀, 제9호 태풍 마
이삭 경로는 한국이 가장 정확

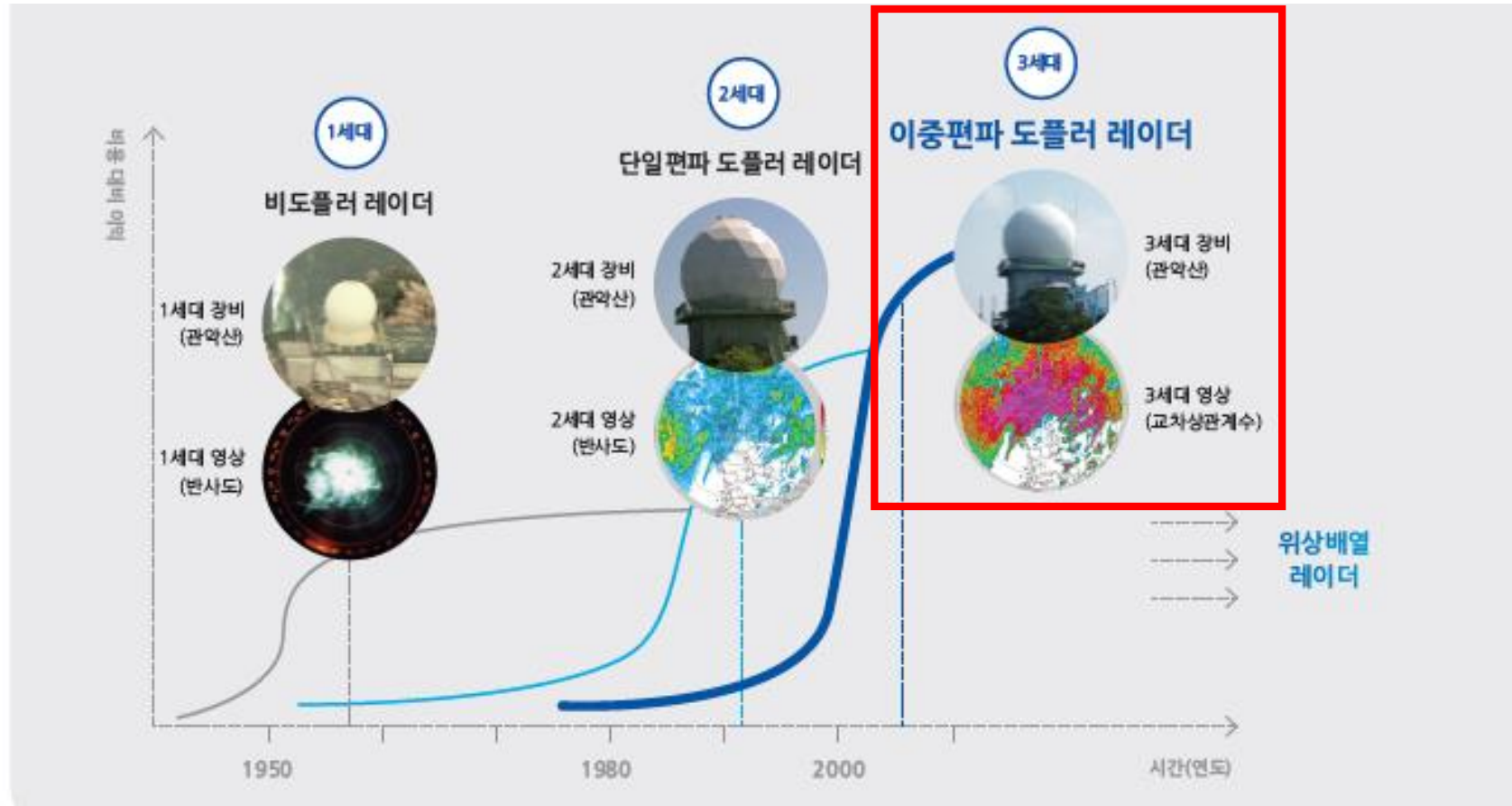
- 강우예측 국내외 현황 : AI 심층학습(딥러닝) 기반 연구 확대



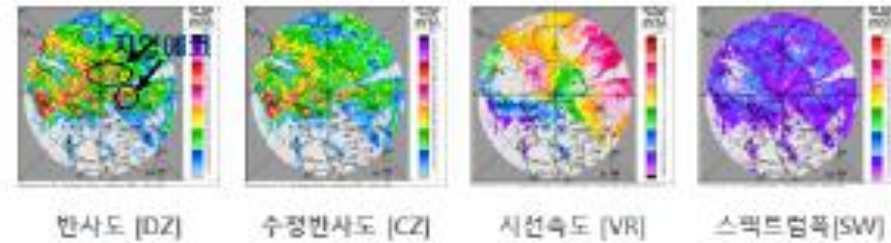
홍콩 기상레이더와 ConvLSTM을 활용한
강우예측 사례 (Shi et al.,2015)

독일 기상레이더와 U-NET을 활용한
강우예측 사례 (Ayzel et al.,2020)

도메인 분석 기상 레이더 기술 변화 동향



- 다중 채널 / 다변량 데이터
- 다양한 시공간적 규모의 데이터
- 유동적인 자료를 정밀하게 체크함
- Vision 관련 분야 중 특히 데이터 규모가 큼



Weather Radar Center
<http://radar.kma.go.kr/lecture/radar/dataflow.do>

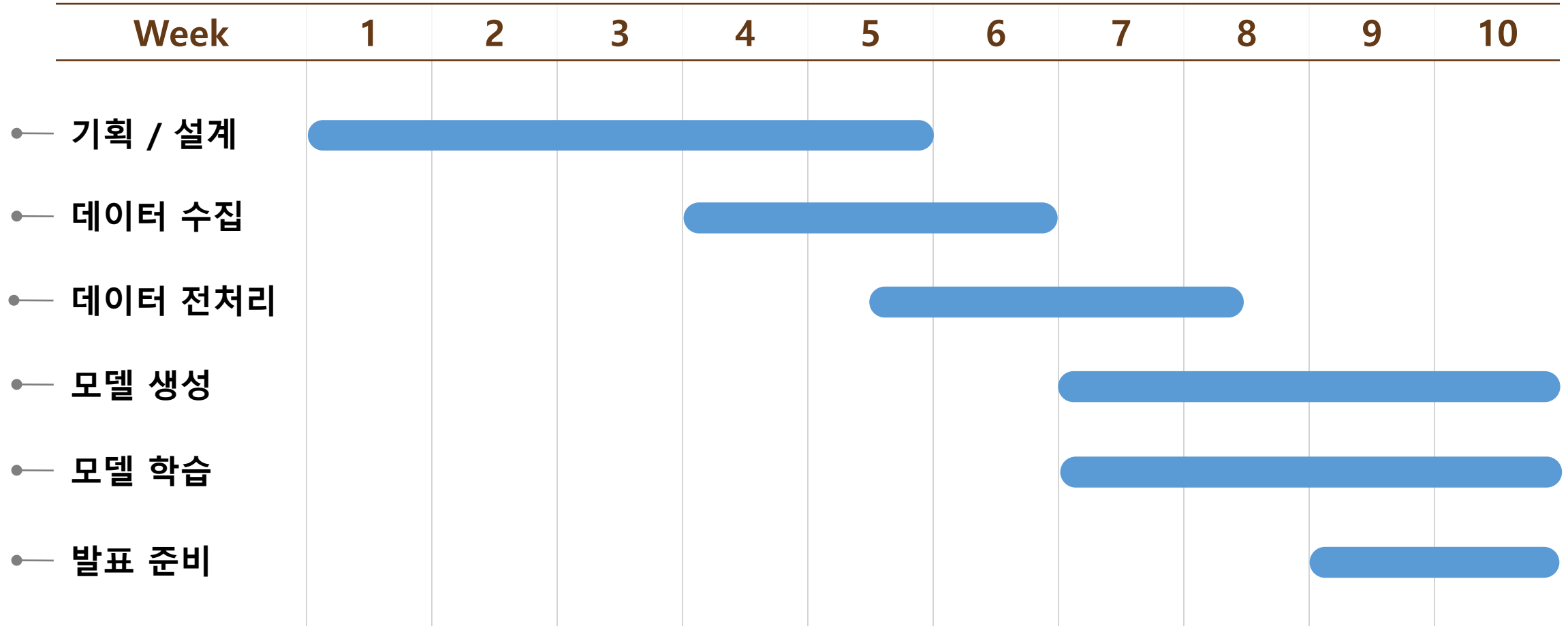
○ RADAR (Radio Detection And Range)

목표물을 향해 전파를 발사한 후 되돌아온 전파(echo)를 분석하여
그 목표물의 여러가지 특성을 조사하는 전파 장치

○ 기상레이더

발사된 전자기파가 비, 눈, 우박 등에 부딪혀 되돌아오는 반사 신호를 분석
→ 강수구름 위치 및 이동상태 추적

- 이미지 픽셀값에 일반적으로 층운형 호우에 쓰이는 식을 적용해 강우량으로 변환 가능
- $$\text{dBZ} = (((\text{pixel} - 0.5) / 255.) * 70) - 10$$
$$Z = \text{pow}(10.0, \text{dBZ} / 10.0) \quad \text{mm}^6/\text{m}^3$$
$$R = \text{pow}(Z / 200.0, 1.0 / 1.6) \quad \text{mm/hr}$$
- dBZ : 1m³ 단위부피내 직경 1mm 물방울 1개 기준 반사도 단위
반사도(Z) : 단위부피내 직경 1mm 물방울 개수 (= 강우강도)





개발 계획 마인드맵

파이썬 통합 개발 환경

개발 플랫폼

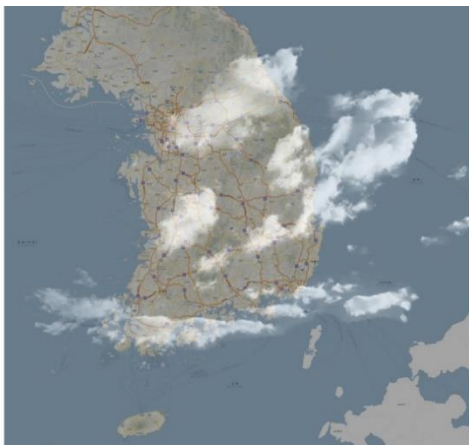


과거 시간별 기상 레이더 구름 반사도 이미지

미래 예측



30분 전



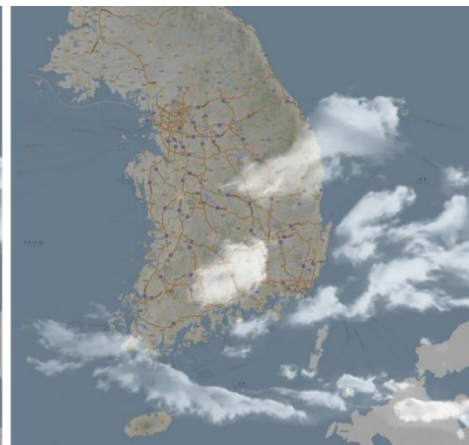
20분 전



10분 전



현재



10분 후

- 데이터셋 수 : 62735 sets
- 각 120x120픽셀 .npy형식 (이미지 5개로 구성)

2010년~2017년 4~10월 강우 사례 레이더 반사도(dBZ)
이미지 픽셀값으로 변환 (0~255 범위 값 그레이스케일)

[출처] 데이콘 '공공데이터 활용 수력 댐 강우예측 AI 경진대회'

- 구글 Colab 환경 구축
- 학습 데이터 압축풀기
- 학습 데이터 로드



```
drive.mount('/content/drive')

Mounted at /content/drive

path = '/content/drive/My Drive/final_prj'

zip_file = zipfile.ZipFile(path+'/train.zip')
zip_file.extractall('.')

# 구글드라이브 train data 파일 경로 지정
train_files = sorted(glob.glob('/content/train/*.npz'))
train_files = np.array(train_files) # [:1000]

# C드라이브 train data 파일 경로 지정
# train_files = glob.glob('C:\\AIP\\rainy_project\\venv\\data\\train/*.npz')
# train_files = np.array(train_files[:1000])

print(len(train_files))
print(train_files[1])

62735
/content/train/train_00001.npz
```

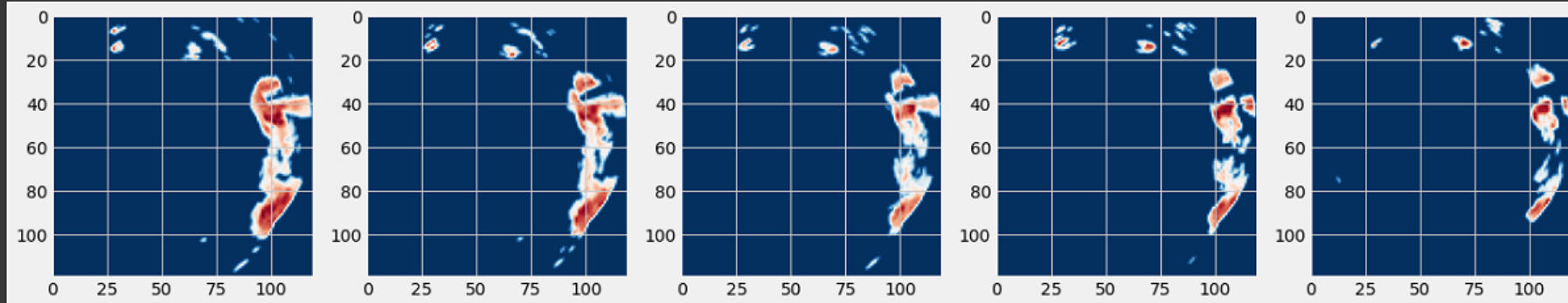
○ .imshow()

```
[ ] color_map = plt.cm.get_cmap('RdBu')
    color_map = color_map.reversed()
    image_sample = np.load(train_files[42])

[ ] plt.style.use('fivethirtyeight')
    plt.figure(figsize=(20, 20))

    for i in range(4):
        plt.subplot(1,5,i+1)
        plt.imshow(image_sample[:, :, i], cmap=color_map)

    plt.subplot(1,5,5)
    plt.imshow(image_sample[:, :, -1], cmap = color_map)
    plt.show()
```



- 결측치/이상치 제거
- Generator 함수 정의
 - Feature(X), Label(Y) 구분
- Transpose, .reshape() → ConvLSTM

```
def trainGenerator():  
    for file in tr_file:  
        dataset = np.load(file)  
        target = dataset[:, :, -1].reshape(120, 120, 1)  
        feature = dataset[:, :, :4]  
  
        yield (feature, target)  
  
def valGenerator():  
    for file in val_file:  
        dataset = np.load(file)  
        target = dataset[:, :, -1].reshape(120, 120, 1)  
        feature = dataset[:, :, :4]  
  
        yield (feature, target)
```

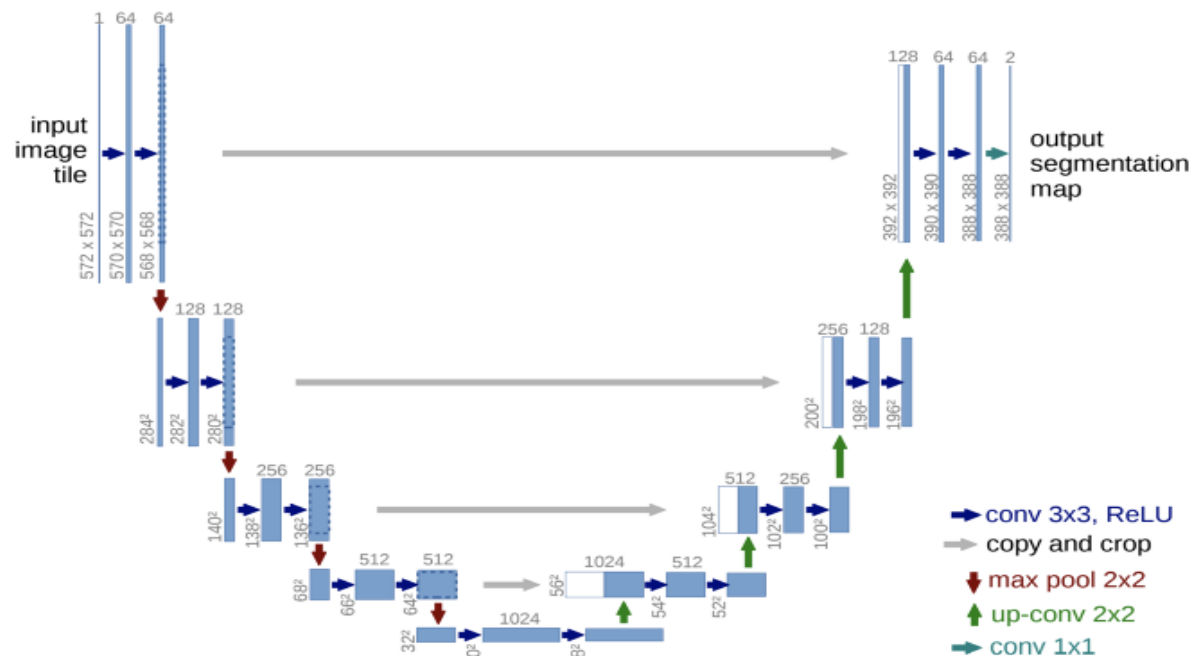
```
def trainGenerator():  
    for file in tr_file:  
        dataset = np.array([np.load(file)]).T  
        Y = dataset[-1, :, :, :].reshape(1, 120, 120, 1)  
        X = dataset[:4, :, :, :].reshape(4, 120, 120, 1)  
        yield (X, Y)  
  
def valGenerator():  
    for file in val_file:  
        dataset = np.array([np.load(file)]).T  
        Y = dataset[-1, :, :, :].reshape(1, 120, 120, 1)  
        X = dataset[:4, :, :, :].reshape(4, 120, 120, 1)  
        yield (X, Y)
```

○ 기존 모델 사용

- CNN2d (이미지)
- ConvLSTM (이미지+시계열)
- Unet (upsampling+concat)

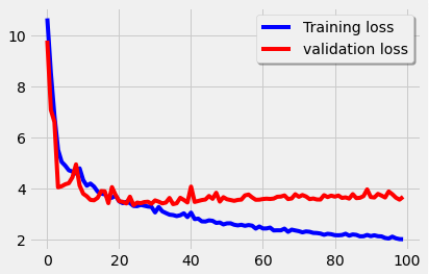
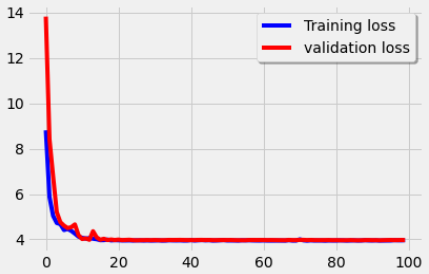
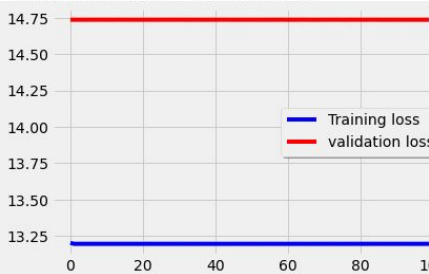
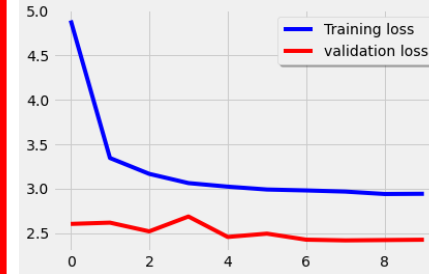
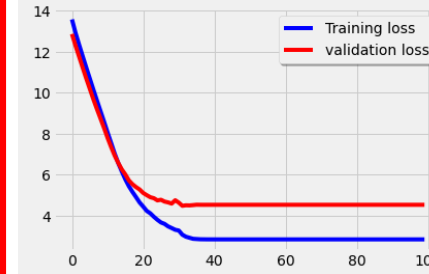
○ 전이학습 모델 생성

- CNN2d+Unet
- ConvLSTM+Unet



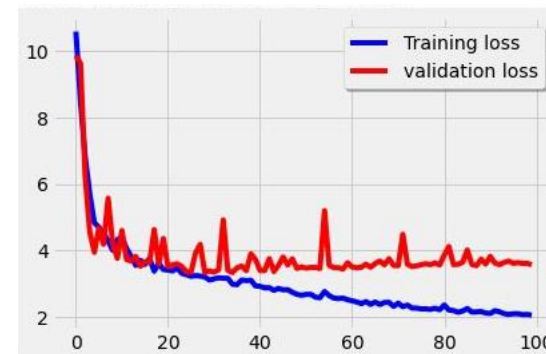
- Unet은 기존 의료계에서 세포 이미지 인식에 성능이 좋았던 모델
- 세포 이미지의 특징 : 전체 이미지 사이즈 대비 특징적인 부분 비중이 작아 식별이 어려움
 - 구름 이미지 또한 전체 배경 중 구름이 차지하는 부분이 작음
- Unet 형태 모델의 특징
 - 1) 빠른 속도 : Overlap 비율 ↓, 검증된 batch는 생략
 - 2) Localization과 context 동시 인식
 - 3) Upsampling 과정에서 피쳐 채널 수를 증가시켜 학습에 유리
 - 4) Padding을 통해 빈 부분을 채워줌

모델링 예측 모델별 손실 비교

CNN (Kfold cv=5)		ConvLSTM		Unet (Kfold cv=5)		CNN+Unet (Kfold cv=5)		ConvLSTM+Unet	
									
이미지 수	62735	이미지 수	1000	이미지 수	62735	이미지 수	62735	이미지 수	1000
Loss	2.0291	Loss	3.9409	Loss	13.1958	Loss	2.9410	Loss	2.8315
Val_loss	3.6874	Val_loss	3.9481	Val_loss	14.7374	Val_loss	2.4237	Val_loss	4.5201
Epoch	100	Epoch	100	Epoch	100	Epoch	10	Epoch	100
Batch_size	10	Batch_size	8	Batch_size	100	Batch_size	64	Batch_size	8

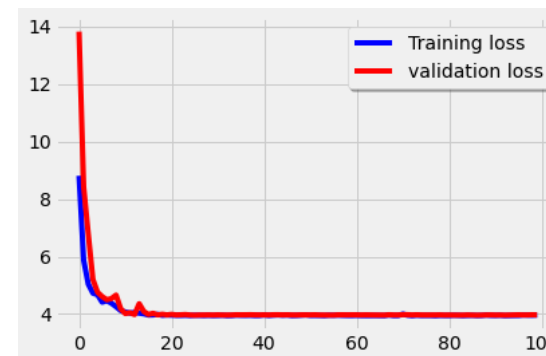
○ CNN

- Epoch 20 구간에서 Overfitting
- Validation loss가 3.5수준에서 더 이상 개선되지 않음



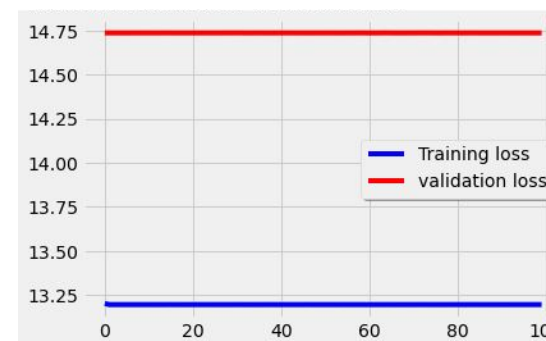
○ ConvLSTM

- 현 개발 환경 상 1000장 이상의 데이터 학습 어려움



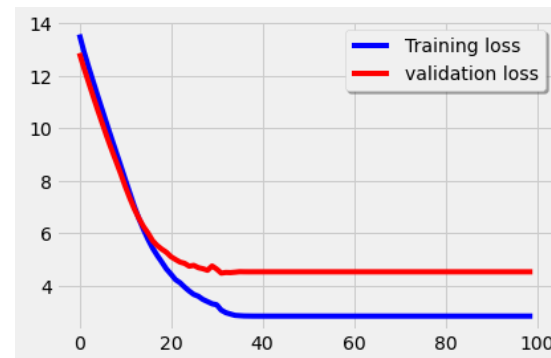
○ Unet

- Train loss와 Validation loss가 10을 넘는 수치 기록
- 학습이 잘 되지 않은 것으로 판단

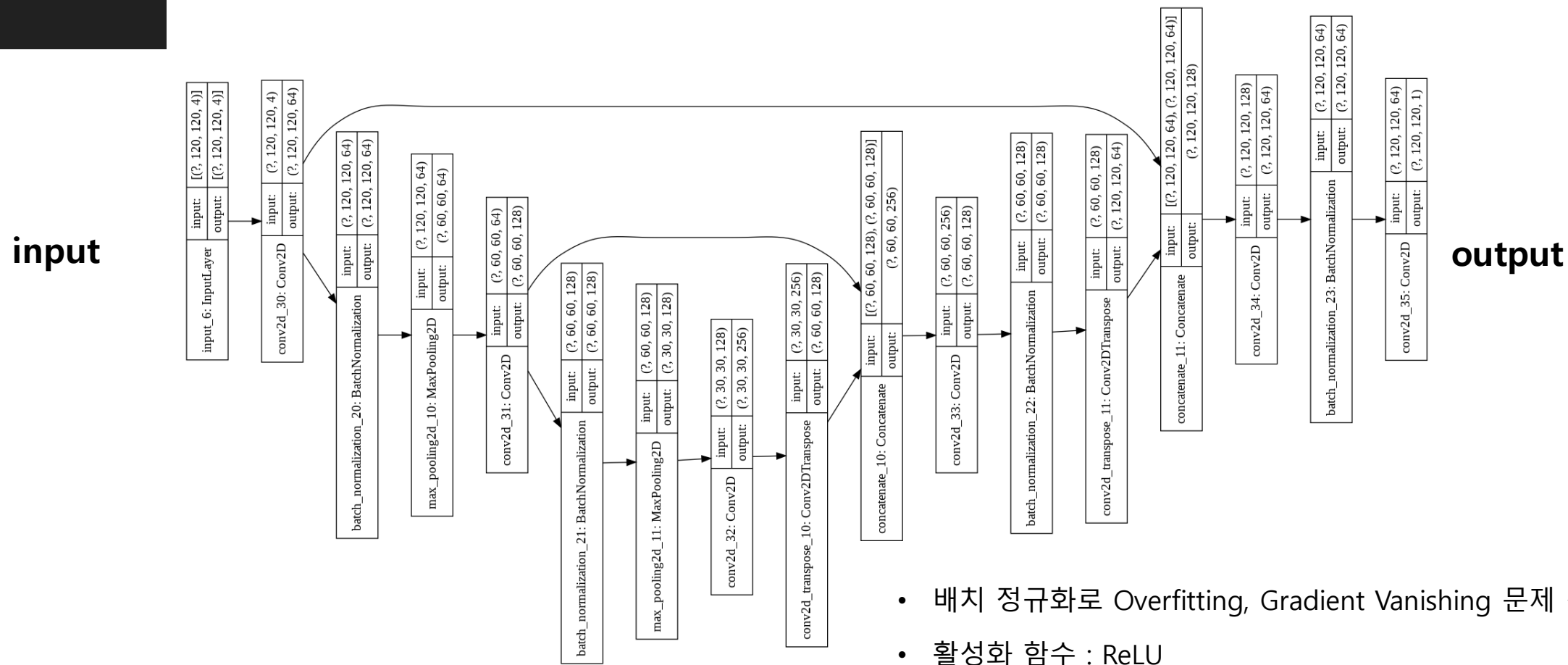


○ ConvLSTM+Unet

- RNN에서 발전된 모델인 LSTM의 특성 상 CNN보다 더 큰 메모리 용량을 필요로 함
→ 전체 학습 데이터셋(67253sets)를 학습 시키기 어려움
- 현 개발환경에서는 Batch Size(8)를 더 이상 키우기 어려움
→ 학습 속도가 현저히 떨어짐



모델링 최종 선정 모델 : CNN+Unet



Input – (Conv – BatchNorm – MaxPooling)*2 – Conv – (ConvT – Concat – Conv – BatchNorm)*2 – **Output**

(120,120,4)

Encoding

Decoding

(120,120,1)

CNN+Unet 모델 Hyper-parameter 튜닝

epoch=100

(early_stop: val_loss)

Layer 추가, batch ↓

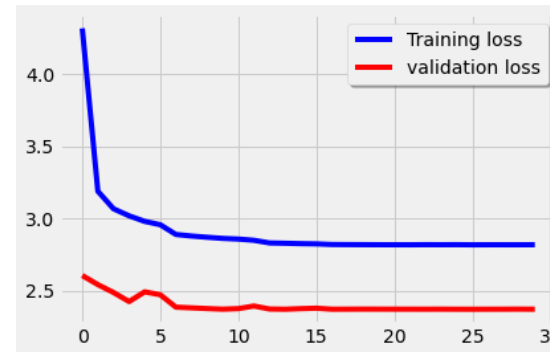
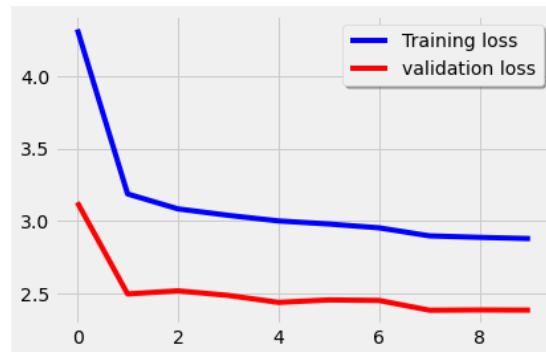
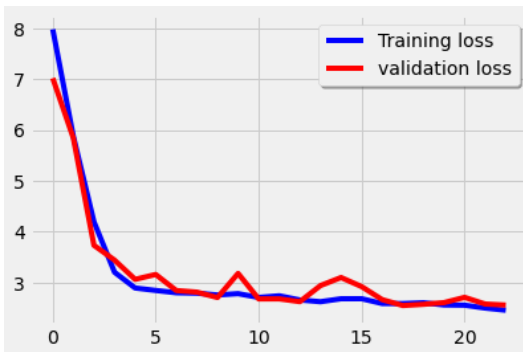
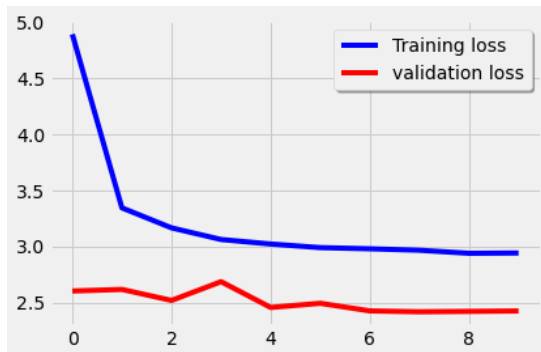
최종 : 1차+2차

기존 모델

1차 튜닝

2차 튜닝

3차 튜닝



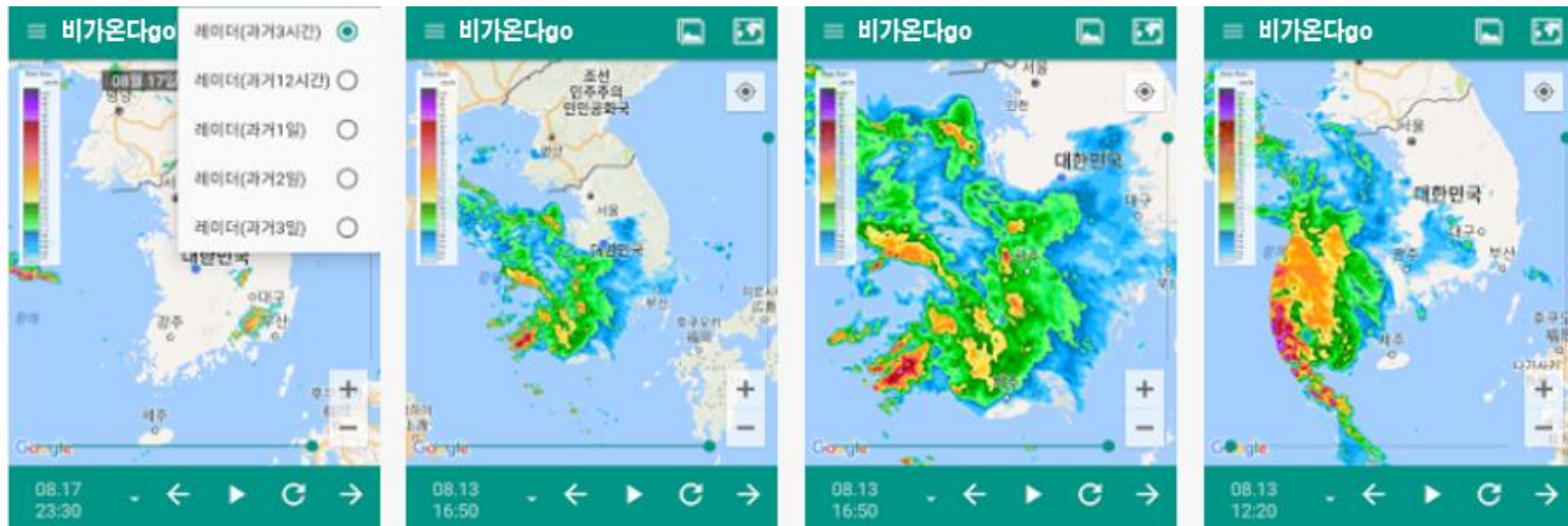
이미지 수	62735	이미지 수	62735	이미지 수	62735	이미지 수	62735
Loss	2.9410	Loss	2.4576	Loss	2.8780	Loss	2.8177
Val_loss	2.4237	Val_loss	2.5586	Val_loss	2.3838	Val_loss	2.3706
Epoch	10	Epoch	22 (early_stop)	Epoch	10	Epoch	30 (early_stop)
Batch_size	64	Batch_size	64	Batch_size	32	Batch_size	32

- 각종 예보에 대한 검증과 평가는 세계기상기구(WMO)에서 권장하는 방법에 따라 실시 (나라마다 차이가 있을 수 있음)
- 강우 예측 검증 시 우리나라는 수치 정확도에 더 무게를 둠 (vs. 미국 : CSI)
- MAE (Mean Absolute Error)
 - 일정 영역 내 관측강수량과 예측강수량의 차이 (절대값) 평균
 - 일반적인 회귀 모델의 경우 MSE를 더 많이 사용
 - 그러나 MSE는 오차를 제곱하기 때문에 오차가 커지면 커질 수록 손실이 제곱으로 커짐
→ 100개 예측 중 99개가 잘 맞아도 1개가 크게 틀리면 그 1개 값의 영향을 크게 받게 됨
 - 기상 예측 모델 평가를 위해서는 MSE보다 강건한 특성을 지닌 MAE를 사용

- 미국에서 주로 사용하는 평가지표 CSI도 평가에 반영
 - CSI (Critical Success Index) :
임계성공지수 = 정확히 예측한 경우 / 강수 발생과 관련된 전체 경우

		Observed		
		Yes	No	Total
Forecast	Yes	Hits	False Alarms	Forecast Yes
	No	Misses	Correct Negatives	Forecast No
	Total	Observed Yes	Observed No	Total

- 구름 모션 예측 결과 이미지와 변환한 강우량 수치를 보여줄 수 있도록 웹 서비스화



Thank You