

# 농업 환경 변화에 따른 작물 병해 진단 AI 경진대회

---



Team 코딩시국코시국  
Member 신유승, 한지용, 박성욱

# CONTENT

## 01 Data Preprocessing

---

- Image Data
- Tabular Data

## 02 Model

---

- CNN2RNN
- CNN Model
- RNN Model

## 03 Experiments

---

- TTA
- Bidirectional LSTM

## 04 Conclusion

---

- Final Result

INTRO

코딩시국코딩시국  
TEAM

## “Focus Image 활용한 모델 구축 개발”

팀원 신유승, 한지용, 박성욱

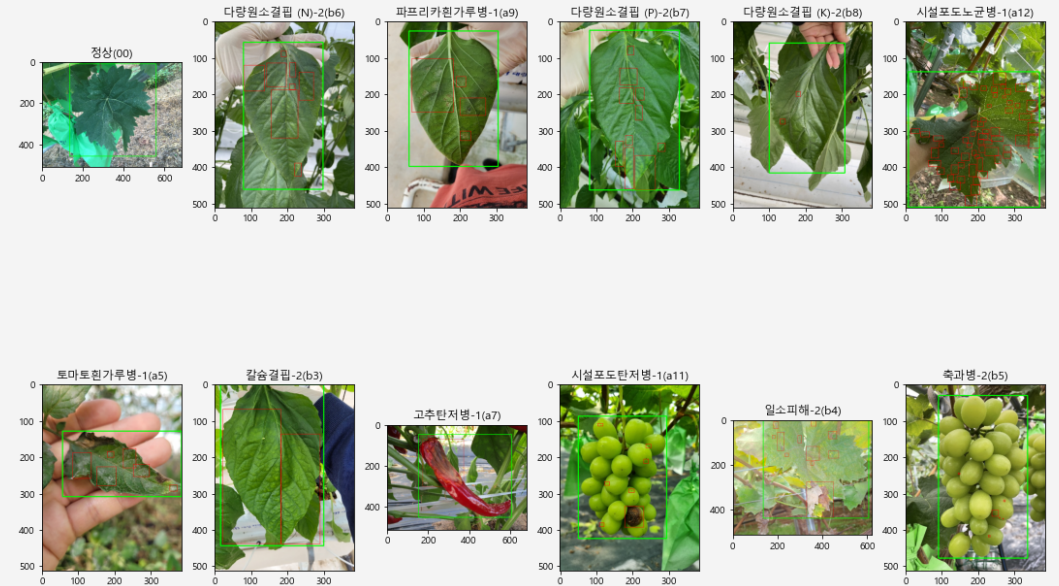
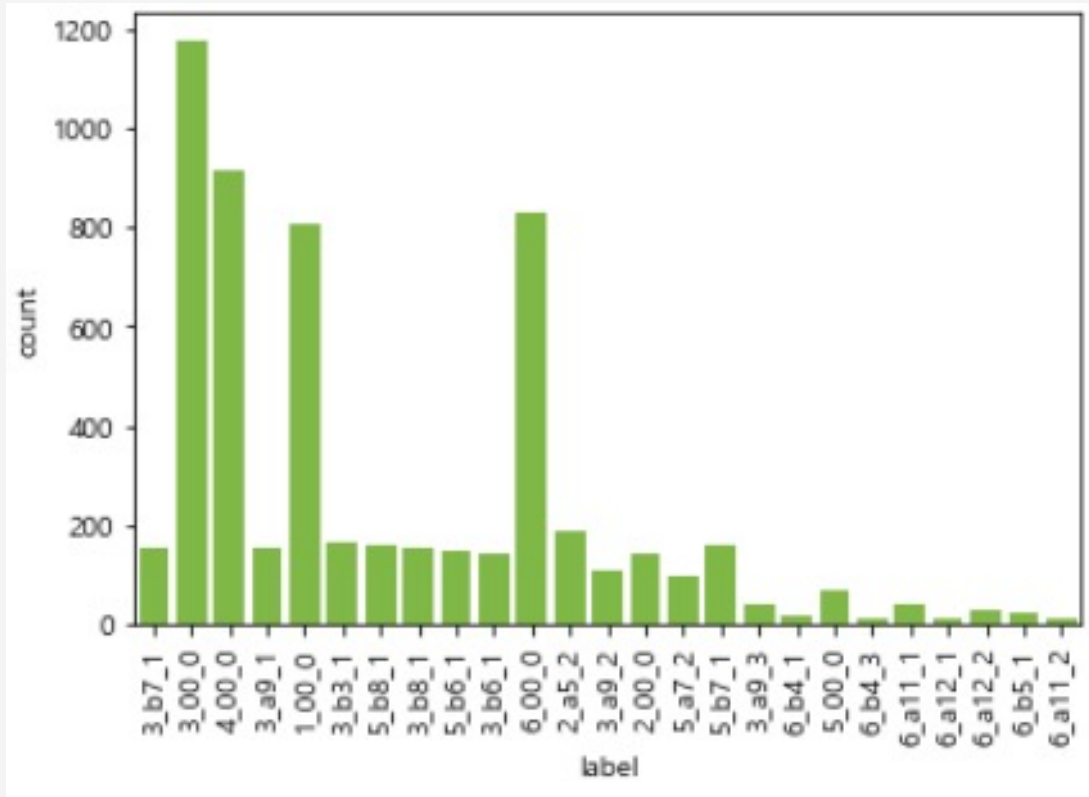
신유승 : 모델 실험 및 자료 제작

한지용 : 모델 전처리 및 개발

박성욱 : 모델 실험 및 자료 리서치

# 01 Data Preprocessing

## Image Data



- 총 111개 Class 중 학습 데이터에 분포하는 Class는 총 25개
- 특정 Class의 이미지가 거의 없기 때문에 Data Augmentation이 필요

# 01 Data Preprocessing

## Image Data (Focus Image)

Original



Focus Crop



## Focus Image 기법 적용

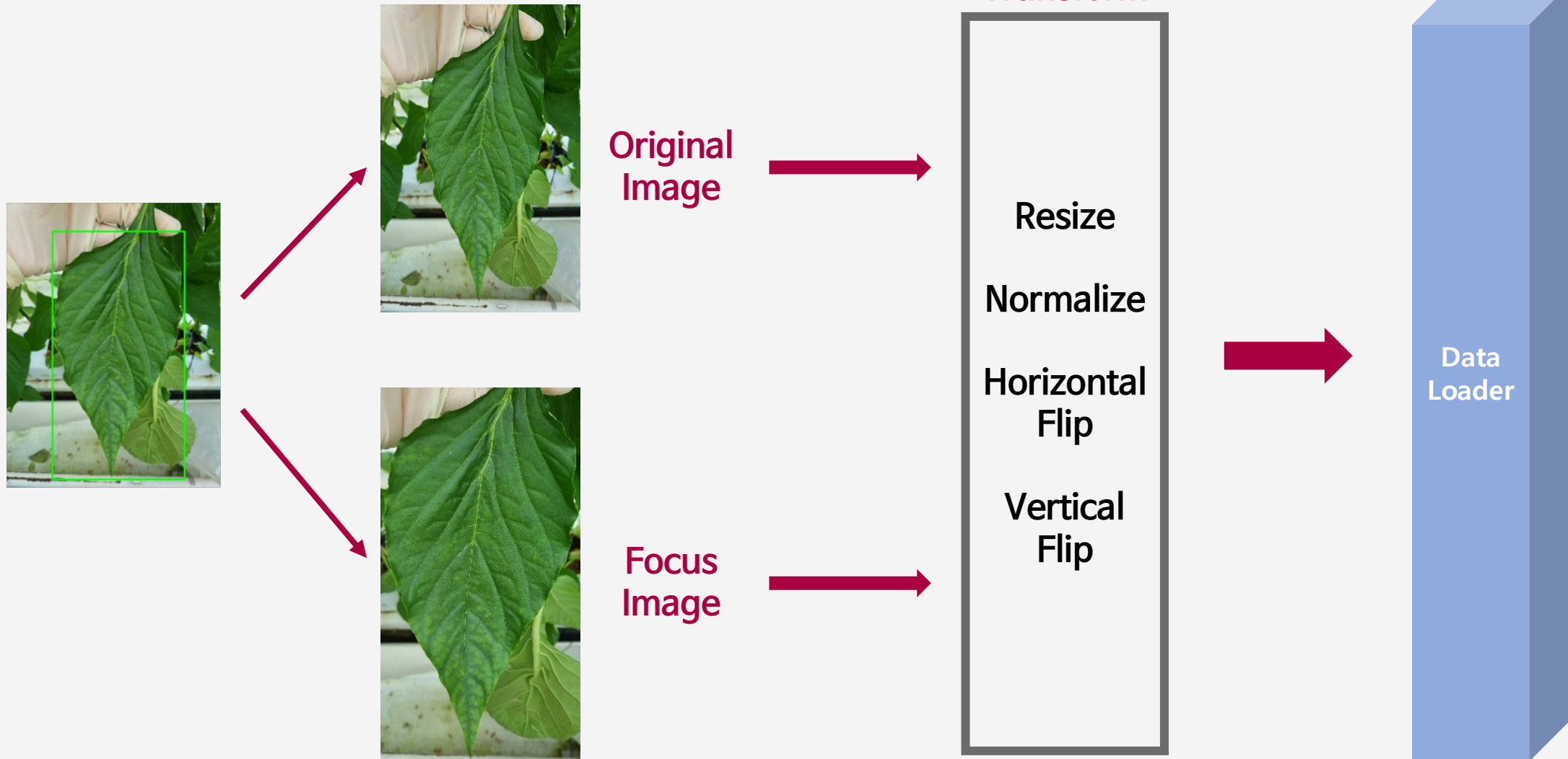
객체의 Bounding Box를 통해 이미지를 따로 저장하여 학습에 중요도가 떨어지는 손, 다른 줄기 등의 불필요한 요인들을 제외하여 가장 중요한 특징 값만을 추출한 이미지 생성함



Focus Image Sample

# 01 Data Preprocessing

Image Data  
(Data Augmentation)



# 01 Data Preprocessing

## Tabular Data

	측정시간	내부 온도 1 평균	내부 온도 1 최고	내부 온도 1 최저	내부 온도 2 평균	내부 온도 2 최고	내부 온도 2 최저	내부 온도 3 평균	내부 온도 3 최고	내부 온도 3 최저	...	배지 중량 최저	양액 온도 평균	양액 온도 최고	양액 온도 최저	외부 풍향 수치	외부 풍향	외부 빛물 시간	외부 누적일사 평균	양액 금액 누적	양액 배액 누적
0	2021-08-13 15:10:00	28.0	28.1	27.9	-	-	-	-	-	-	...	-	-	-	-	-	-	-	91.1	-	-
1	2021-08-13 15:00:00	28.2	28.3	28.1	-	-	-	-	-	-	...	-	-	-	-	-	-	-	97.5	-	-
2	2021-08-13 14:50:00	28.5	28.7	28.4	-	-	-	-	-	-	...	-	-	-	-	-	-	-	100.8	-	-
3	2021-08-13 14:40:00	28.9	29.0	28.7	-	-	-	-	-	-	...	-	-	-	-	-	-	-	109.5	-	-
4	2021-08-13 14:30:00	29.1	29.2	29.0	-	-	-	-	-	-	...	-	-	-	-	-	-	-	134.8	-	-
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
289	2021-08-11 15:00:00	31.8	31.9	31.7	-	-	-	-	-	-	...	-	-	-	-	-	-	-	468.2	-	-
290	2021-08-11 14:50:00	31.1	31.2	31.0	-	-	-	-	-	-	...	-	-	-	-	-	-	-	500.1	-	-
291	2021-08-11 14:40:00	31.3	31.3	31.1	-	-	-	-	-	-	...	-	-	-	-	-	-	-	263.1	-	-

## CSV Features

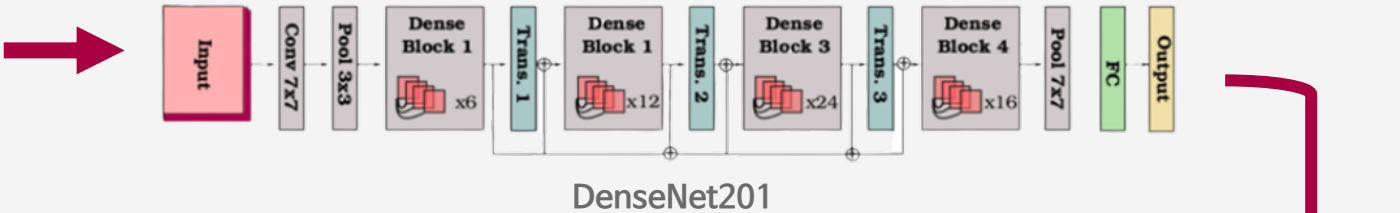
```
# 분석에 사용할 feature 선택
csv_features = ['내부 온도 1 평균', '내부 온도 1 최고', '내부 온도 1 최저', '내부 습도 1 평균', '내부 습도 1 최고',
                '내부 습도 1 최저', '내부 이슬점 평균', '내부 이슬점 최고', '내부 이슬점 최저']
```



# 02 Model

## CNN2RNN

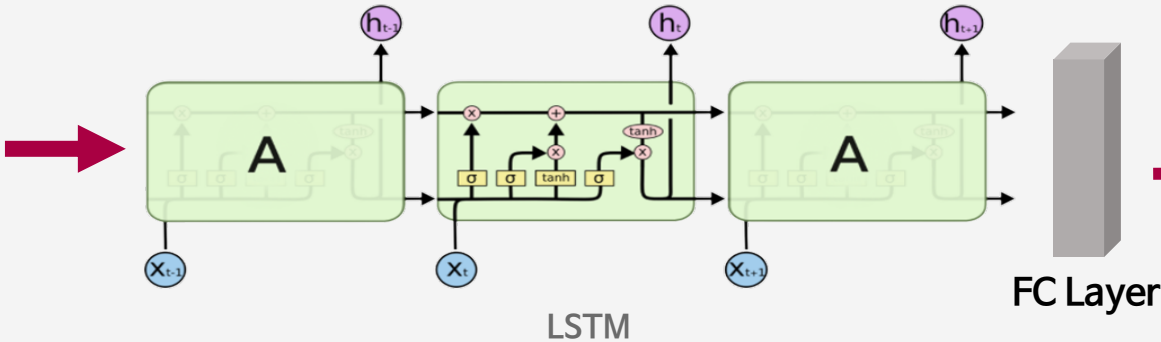
Image Data



DenseNet201

Tabular Data

2021-09-14 7:20	18.5	18.6	18.3
2021-09-14 7:10	18.1	18.3	17.9
2021-09-14 7:00	17.7	17.9	17.5
2021-09-14 6:50	17.3	17.4	17.2
2021-09-14 6:40	17.1	17.3	17.1
2021-09-14 6:30	17	17.1	16.9

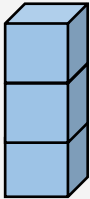


LSTM

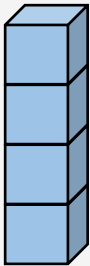
FC Layer



Predict



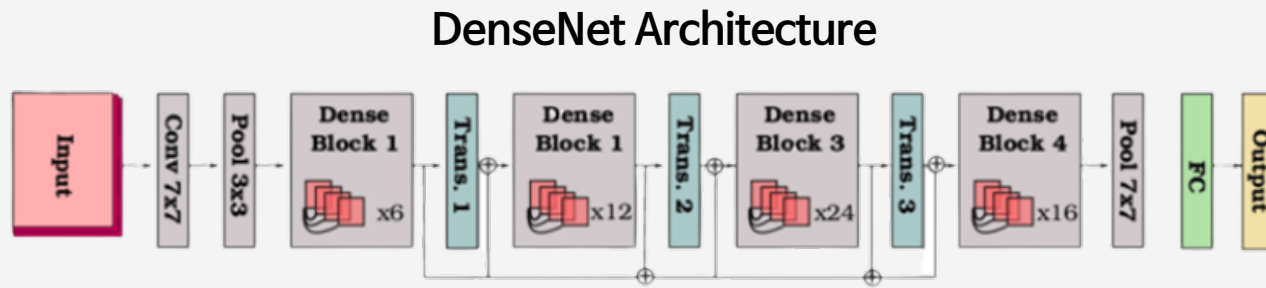
⋮ 111 Class



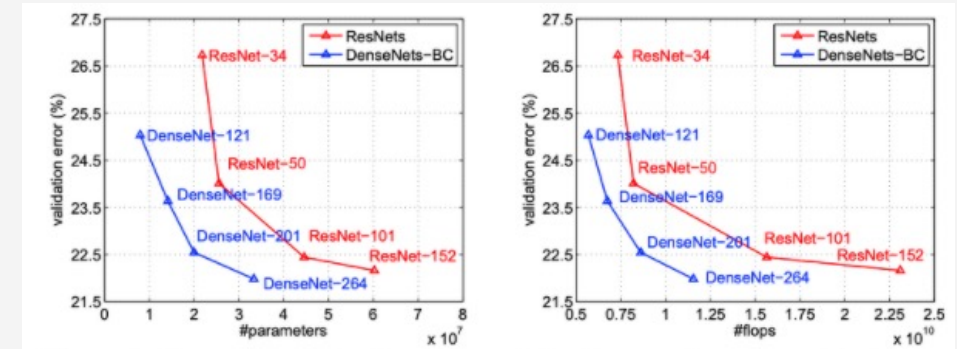


# 02 Model

## CNN Model



Pretrained model (torchvision.models) 사용



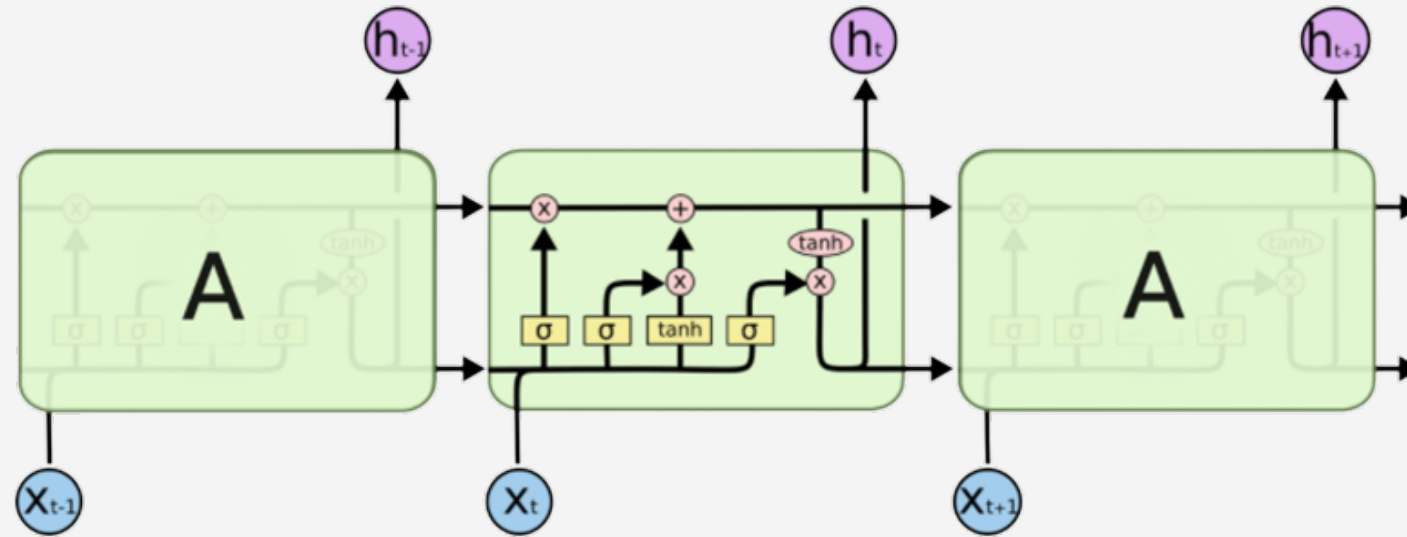
모델 간 Flops 비교

### 모델 선택 이유

- ResNet(Baseline Model) 보다 적은 파라미터
- 적은 파라미터로 높은 성능 (동작시간, 학습의 효율성 고려)
- 다른 모델들 사용 시 최적화의 어려움

## 02 Model

### RNN Model



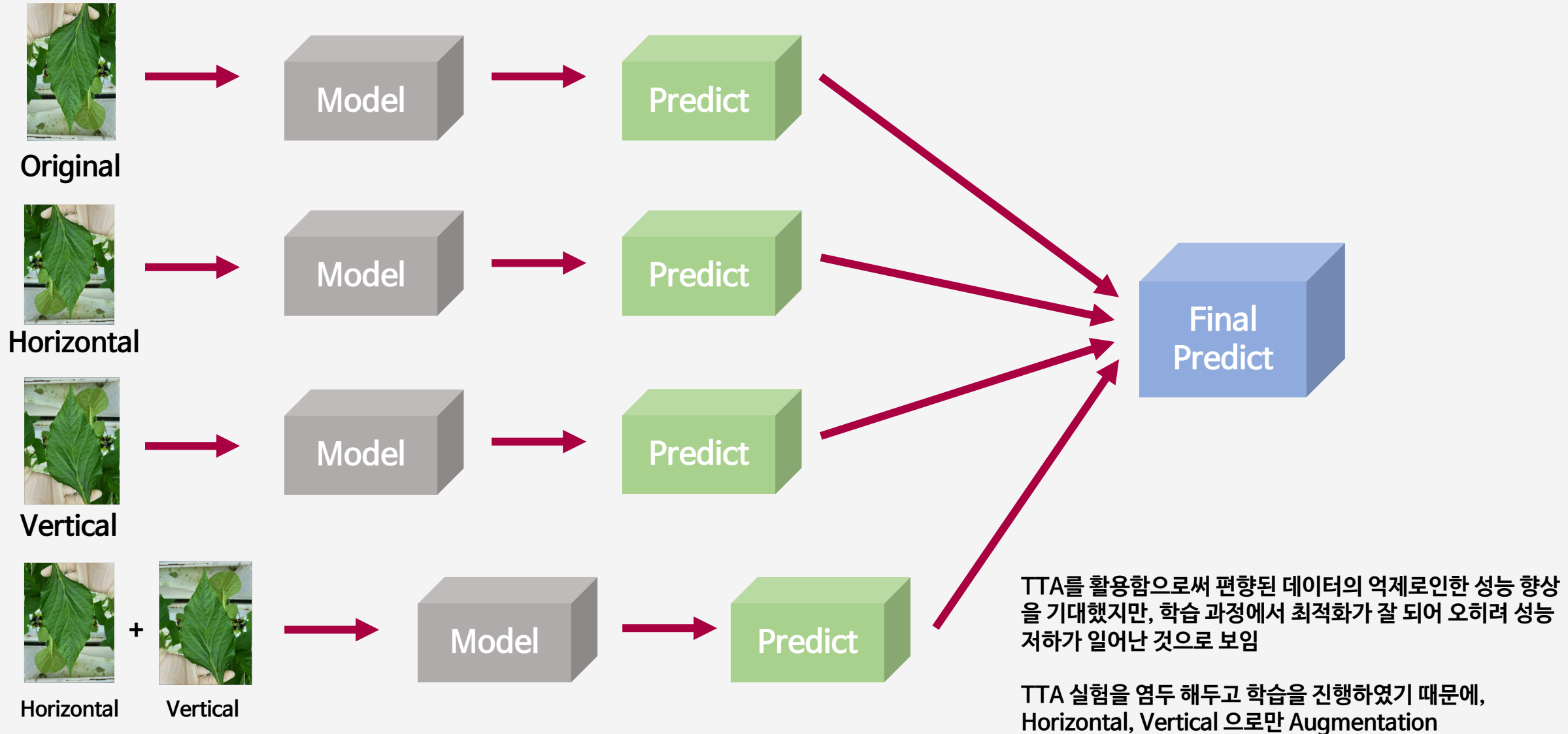
LSTM Architecture

#### 모델 선택 이유

- RNN 모델의 경우 장기 의존성의 문제로 채택하지 않았음
- GRU 모델의 경우 적은 파라미터로 빠른 학습을 할 수 있었지만 그만큼 최적화가 쉽지 않았음

# 03 Experiments

## TTA (Test Time Augmentation)



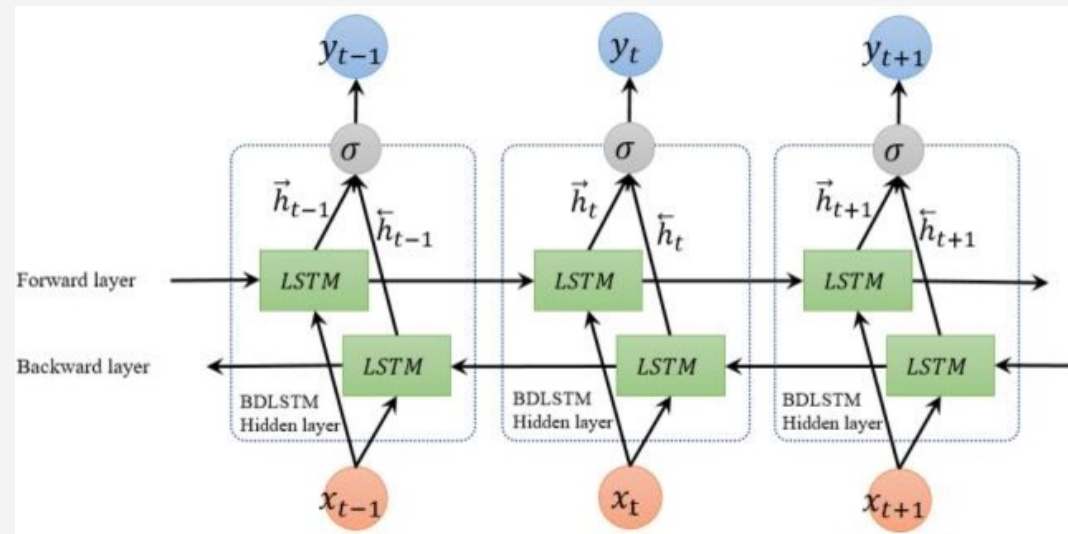
# 03 Experiments

## Bidirectional LSTM

### Bidirectional = True 학습 실험 진행

```
class RNN_Decoder(nn.Module):  
    def __init__(self, max_len, embedding_dim, num_features, class_n, rate):  
        super(RNN_Decoder, self).__init__()  
        self.lstm = nn.LSTM(max_len, embedding_dim, bidirectional=True)  
        self.rnn_fc = nn.Linear(num_features * embedding_dim * 2, 1000)  
        self.pre_final_layer = nn.Linear(1000 + 1000, 512) # densenet out_dim + lstm out_dim  
        self.final_layer = nn.Linear(512, class_n)  
        self.relu = nn.ReLU()  
        self.dropout = nn.Dropout(rate)  
  
    def forward(self, enc_out, dec_inp):  
        # hidden, _ = self.gru(dec_inp)  
        hidden, _ = self.lstm(dec_inp)  
        hidden = hidden.view(hidden.size(0), -1)  
        hidden = self.rnn_fc(hidden)  
        concat = torch.cat([enc_out, hidden], dim=1) # enc_out + hidden  
        fc_input = concat  
        output = self.relu(self.dropout((self.pre_final_layer(fc_input))))  
        output = self.final_layer(output)  
        return output
```

### Bidirectional LSTM Network



## 04 Conclusion

---

### Final Result

#### \* Training Parameter Information

Epoch : 1500

Learning Rate :  $1e-4$

ReduceLROnPlateau patience=30, factor =0.1 로 학습 진행

Early Stopping : Patience 50 (f1 score 기준으로 진행)

Public Leaderboard Score : **0.9502116429**

Private Leaderboard Score : **0.9530437971**

**THANK**

---

**YOU**