AI VIETNAM
Free AI Course 2020

Model Generalization

(Draft)

Quang-Vinh Dinh
Ph.D. in Computer Science

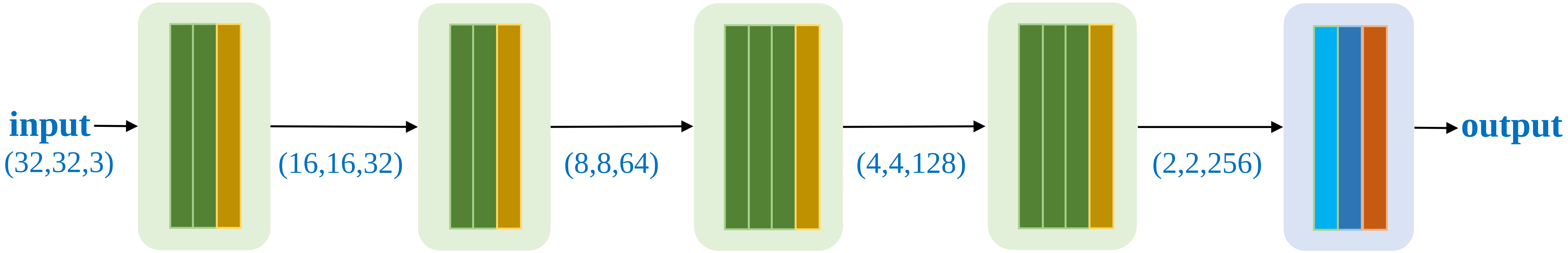
Year 2020

Outline

- Introduction to Numpy
- Numpy Array Indexing
- Numpy Array Operations
- Broadcasting
- Data Processing

Model Generalization

Cifar-10 Dataset



Data Normalization
(convert to 0-mean and 1-deviation)

$$\bar{X} = \frac{X - \mu}{\sigma}$$

$$\mu = \frac{1}{n}\sum_i X_i$$

$$\sigma = \sqrt{\frac{1}{n}\sum_i (X_i - \mu)^2}$$

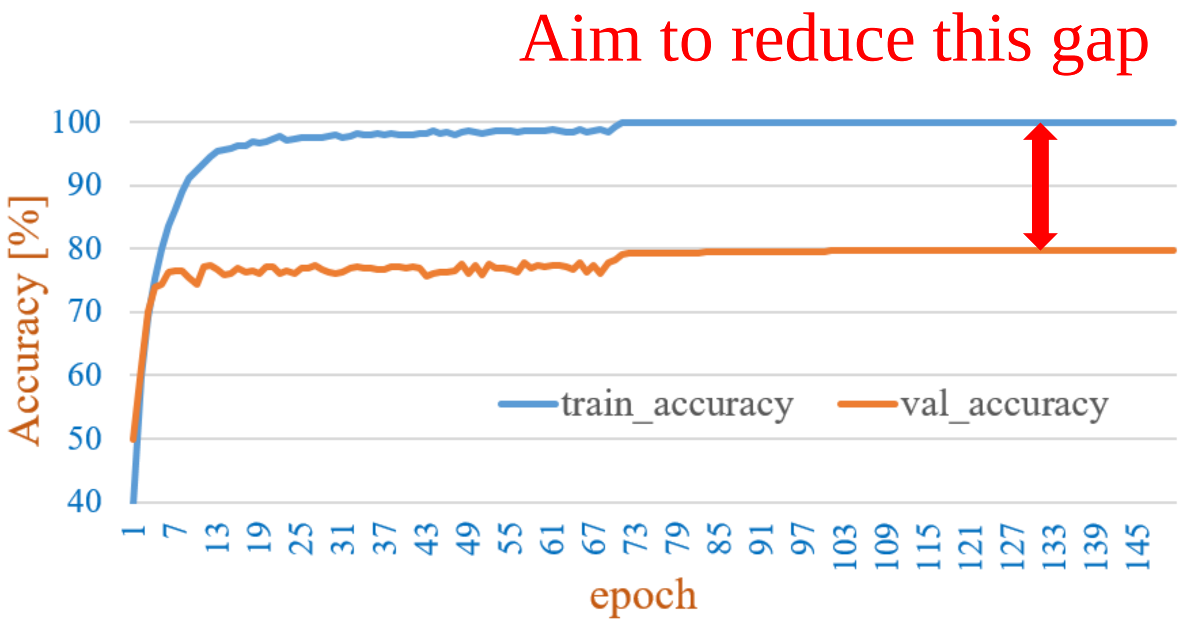
(3x3) Convolution padding='same' stride=1 + ReLU

(2x2) max pooling

Flatten

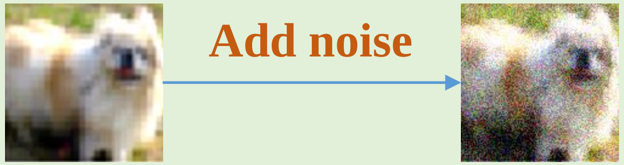
Dense Layer-10 + Softmax

Dense Layer-512 + ReLU



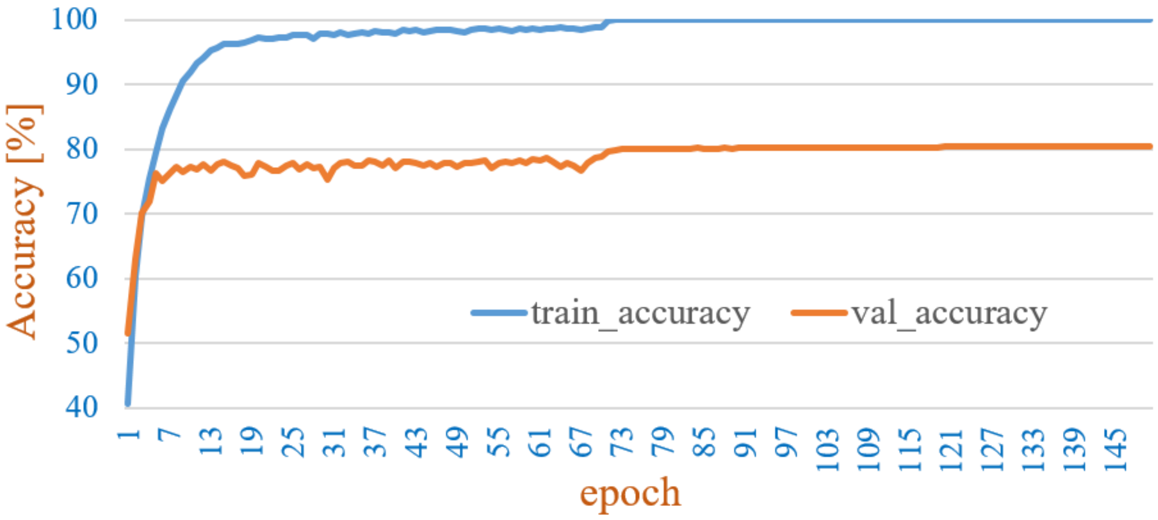
Model Generalization

❖ Trick 1: ‘Learn hard, ’ – randomly add noise to training data



In Keras

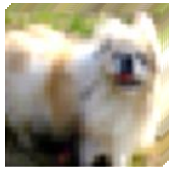
```
if tf.random.uniform(()) > 0.5:
        noise = tf.random.normal((32, 32, 3))/100.0
        image = image+noise

    return image, label
```



val_accuracy increases from ~80.2% to ~80.9%

Model Generalization

❖ Trick 2: Batch normalization



mini-batch 1

mini-batch 2

$$(\mu_1, \sigma_1) \neq (\mu_2, \sigma_2)$$

very likely

Add noise to the output of BN layers

Input data for a node in batch normalization layer

$$X = \{X_1, \dots, X_m\}$$

m is mini-batch size

Compute mean and variance

$$\mu = \frac{1}{m}\sum_{i=1}^{m} X_i \qquad \sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(X_i - \mu)^2$$

Normalize X_i

$$\hat{X}_i = \frac{X_i - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

ϵ is a very small value

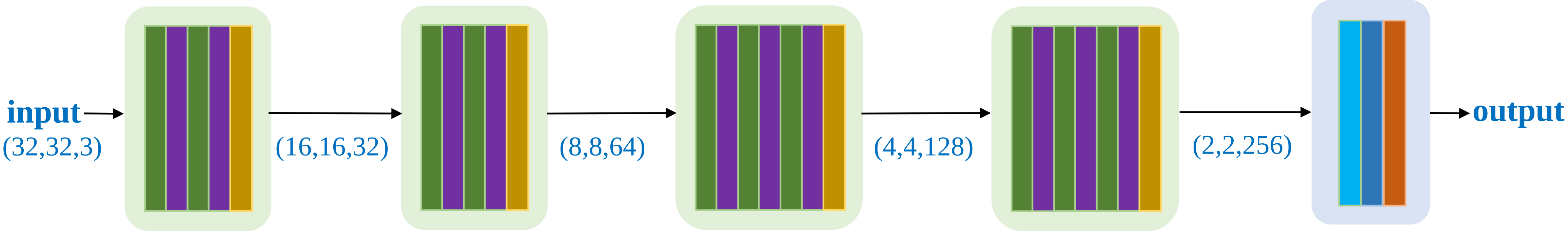
Scale and shift $\hat{X}_i$

$$Y_i = \gamma \hat{X}_i + \beta$$

γ and β are two learning parameters

Model Generalization

❖ Trick 2: Batch normalization



(3x3) Convolution
padding='same'
stride=1 + ReLU

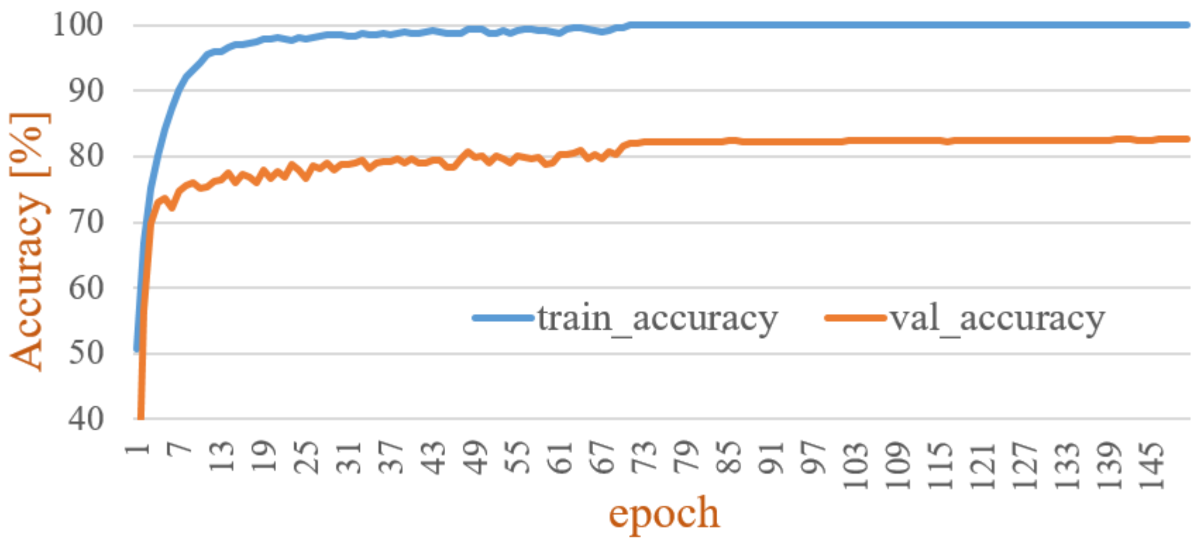
Flatten

(2x2) max pooling

Dense Layer-512
+ ReLU

Batch
normalization

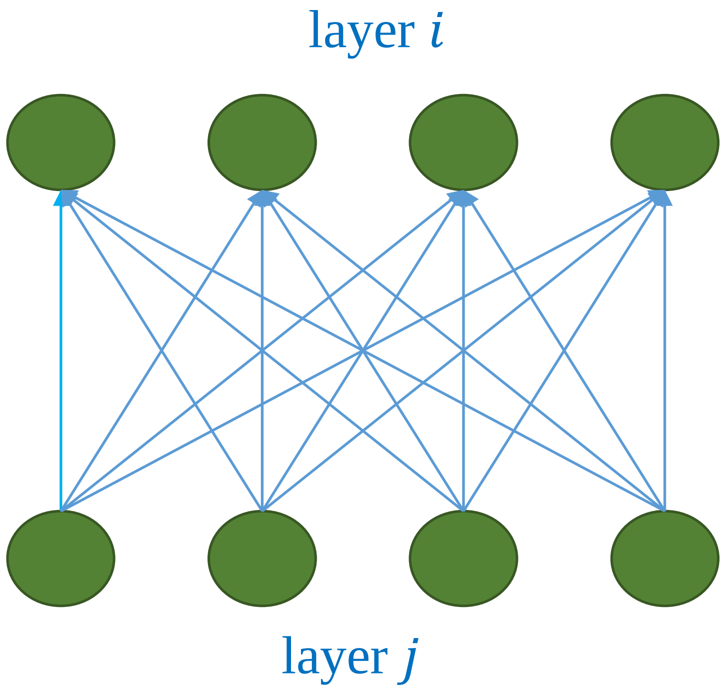
Dense Layer-10
+ Softmax



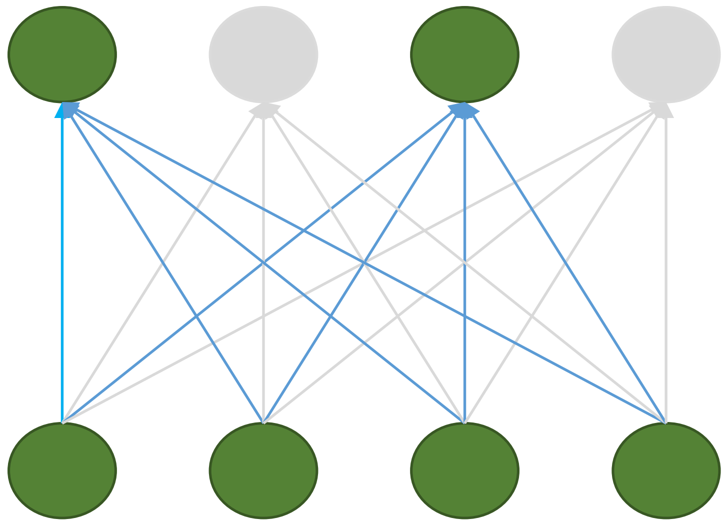
val_accuracy increases from ~80.9% to ~82%

Model Generalization

❖ Trick 3: Dropout



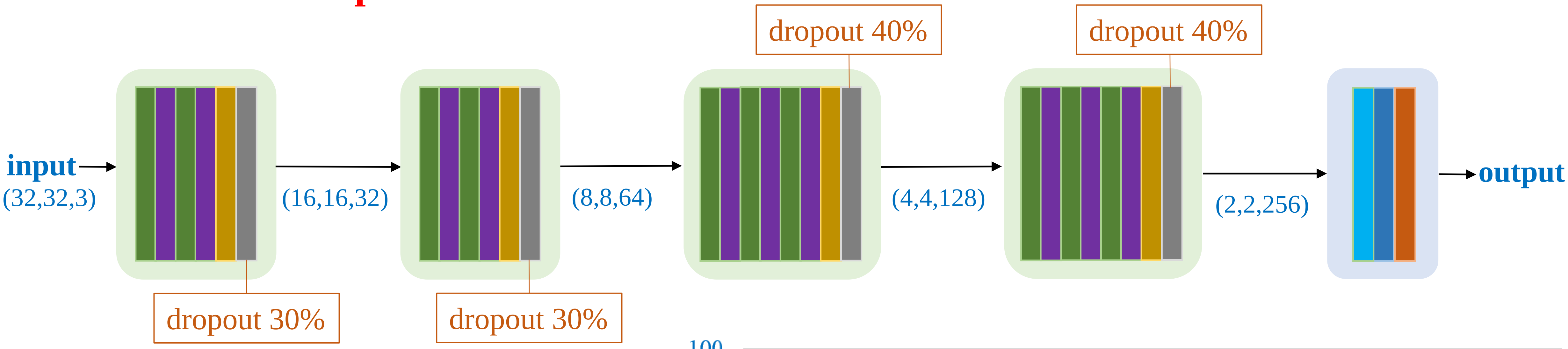
Apply dropout 50% to layer i



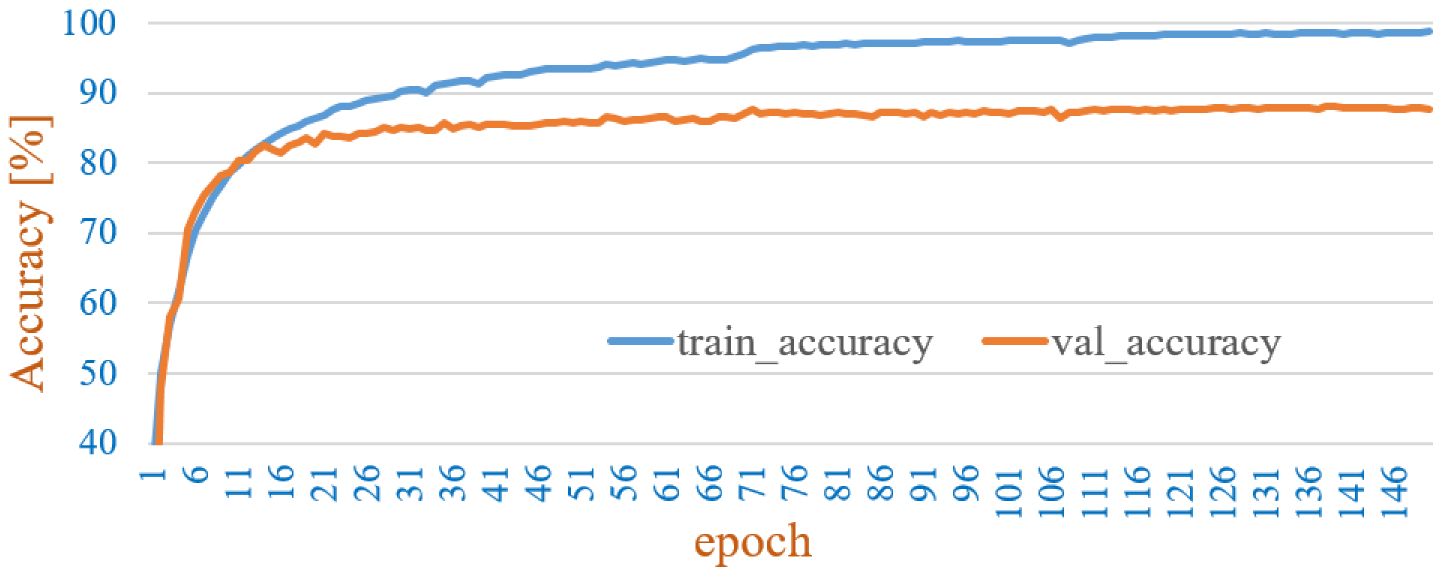
~50% nodes randomly selected in the i^{th} layer are set to zeros (kind of noise adding)

Model Generalization

❖ Trick 3: Dropout



val_accuracy increases from ~82% to ~87.9%



Model Generalization

❖ Trick 4: Kernel regularization

$$L = crossentropy \ + \underbrace{\lambda_1 \|W\|}_{L_1 \text{regularization}} \ + \underbrace{\lambda_2 \|W\|^2}_{L_2 \text{regularization}}$$

Prevent network from focusing on specific features

Smaller weights
→ simpler models

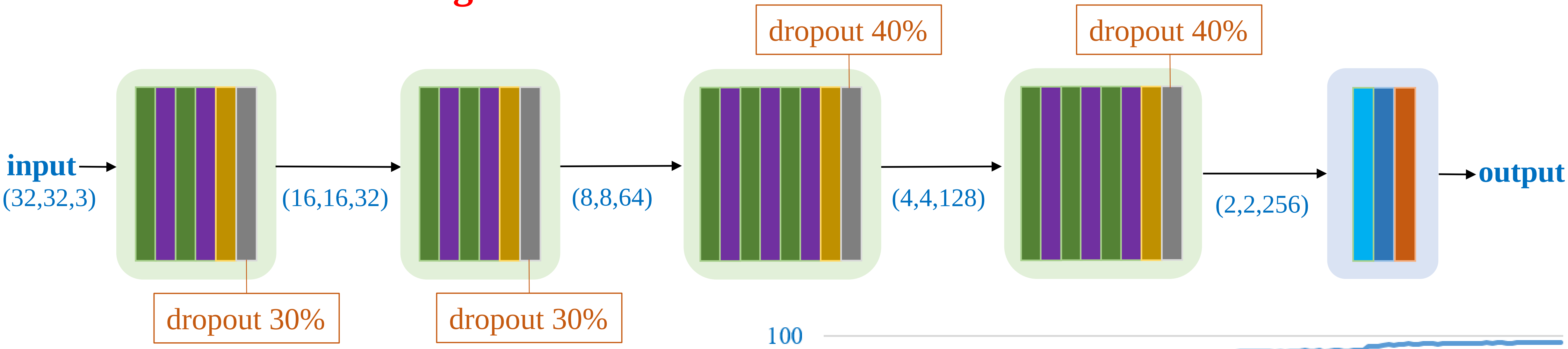
In keras

```
Conv2D(32, (3,3), padding='same', activation='relu',
       kernel_regularizer=regularizers.l1_l2(decay1,decay2))
```

AI VIETNAM
Free AI Course

Model Generalization

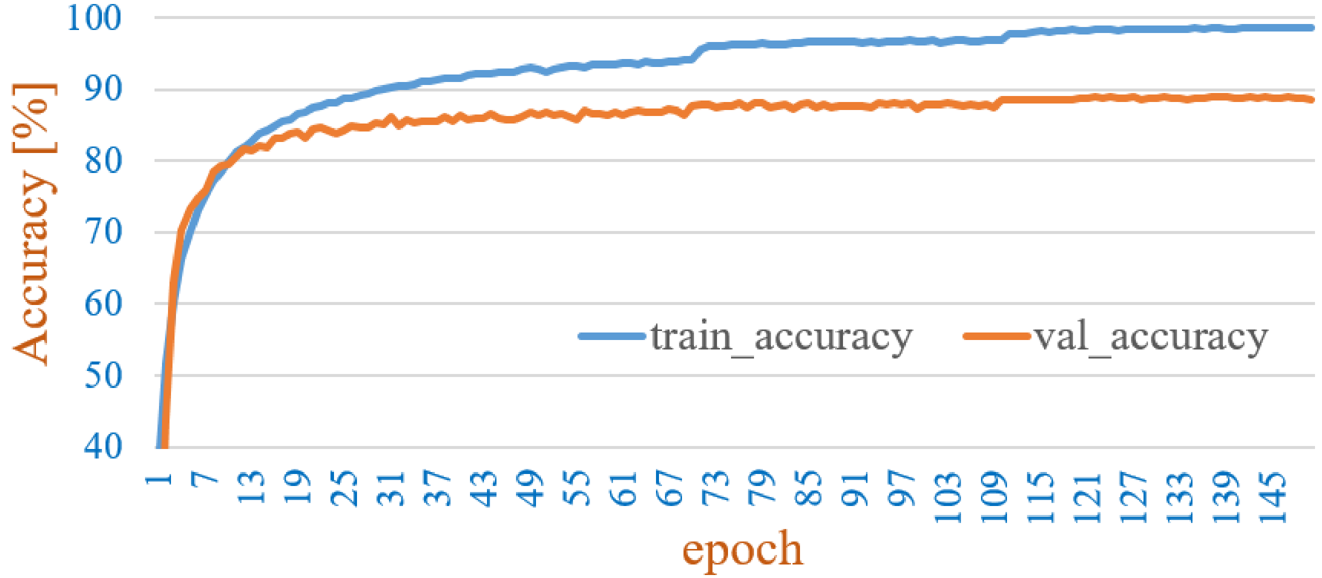
❖ Trick 4: Kernel regularizer



Dropout

(3x3) Convolution
padding='same'
stride=1 + ReLU +
kernel regularization

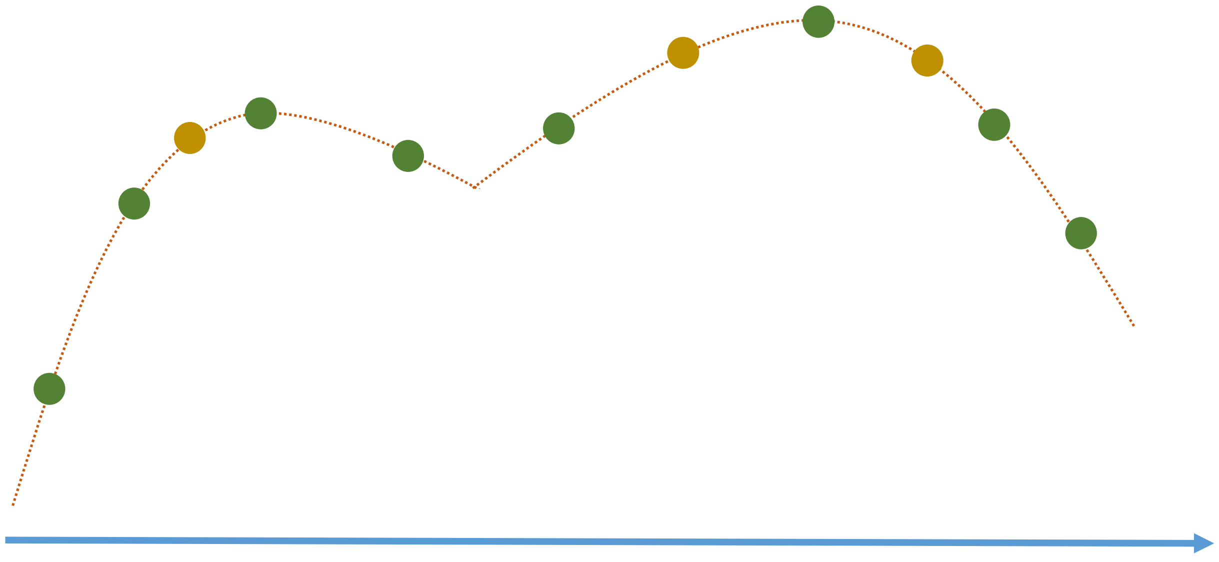
val_accuracy increases from ~87.9% to ~88.9%



Model Generalization

❖ Trick 5: Data augmentation

A normal case

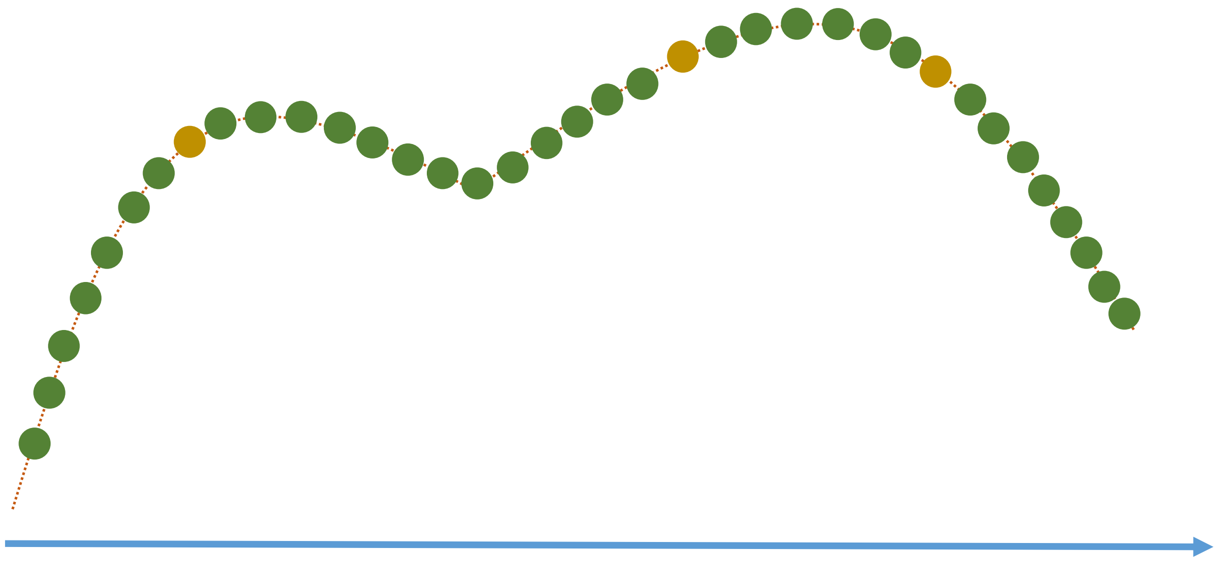


Data distribution

Testing data

Training data

A perfect case: Have unlimited training



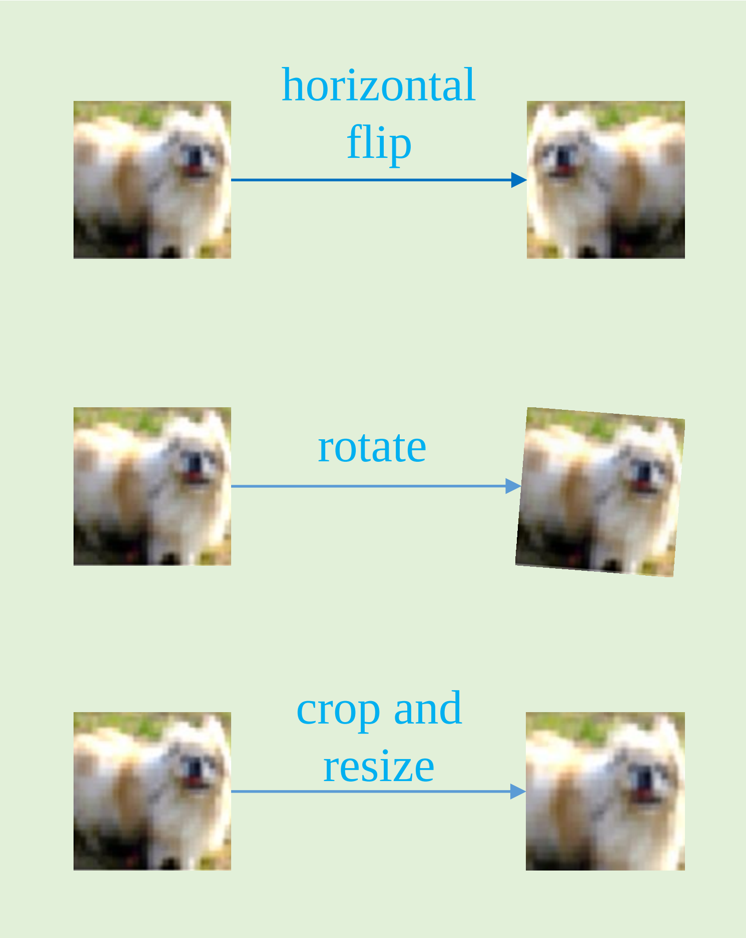
Training data cover the whole distribution

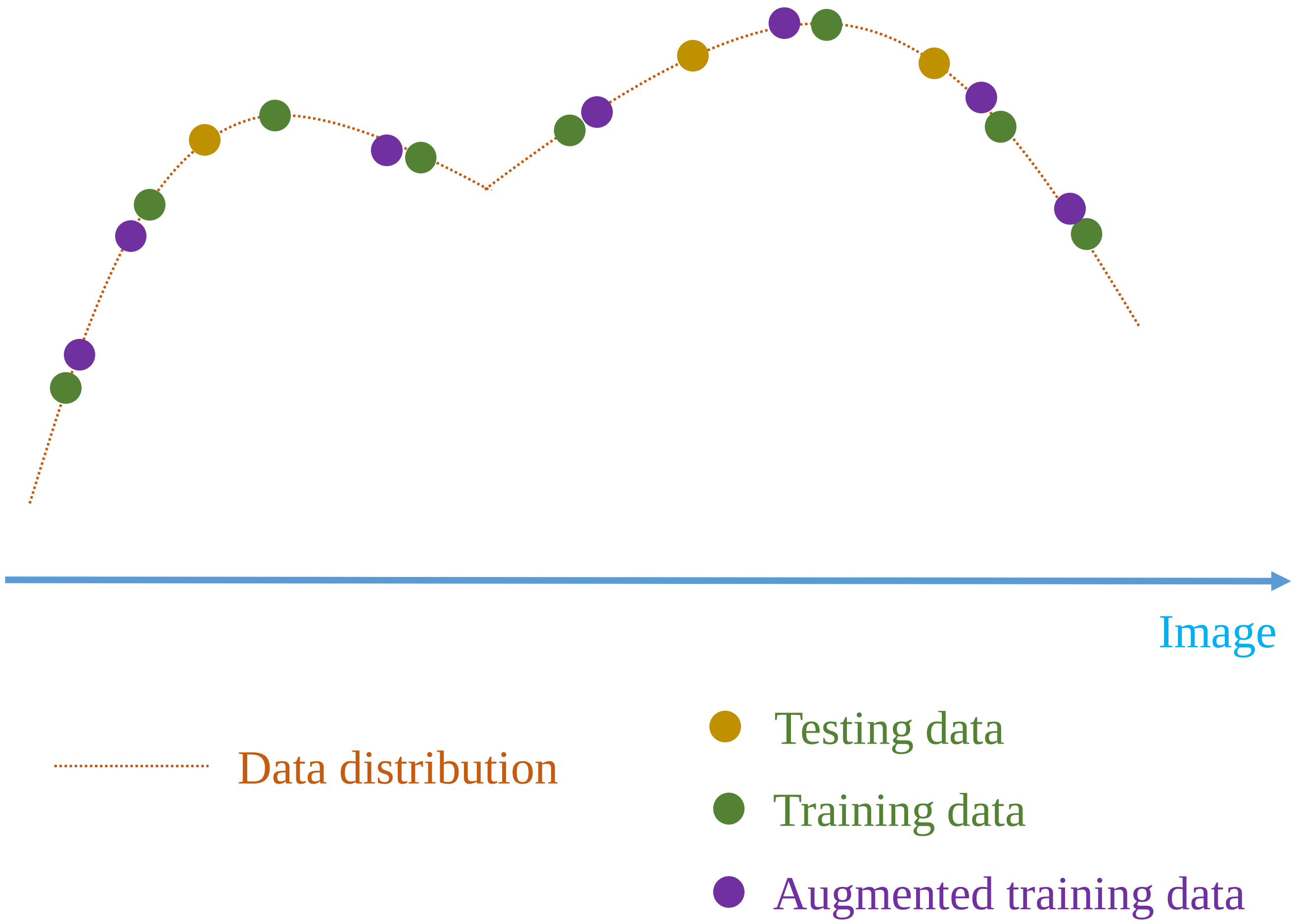
But, impractical!!!

Model Generalization

❖ Trick 5: Data augmentation

Increase data by altering the training data

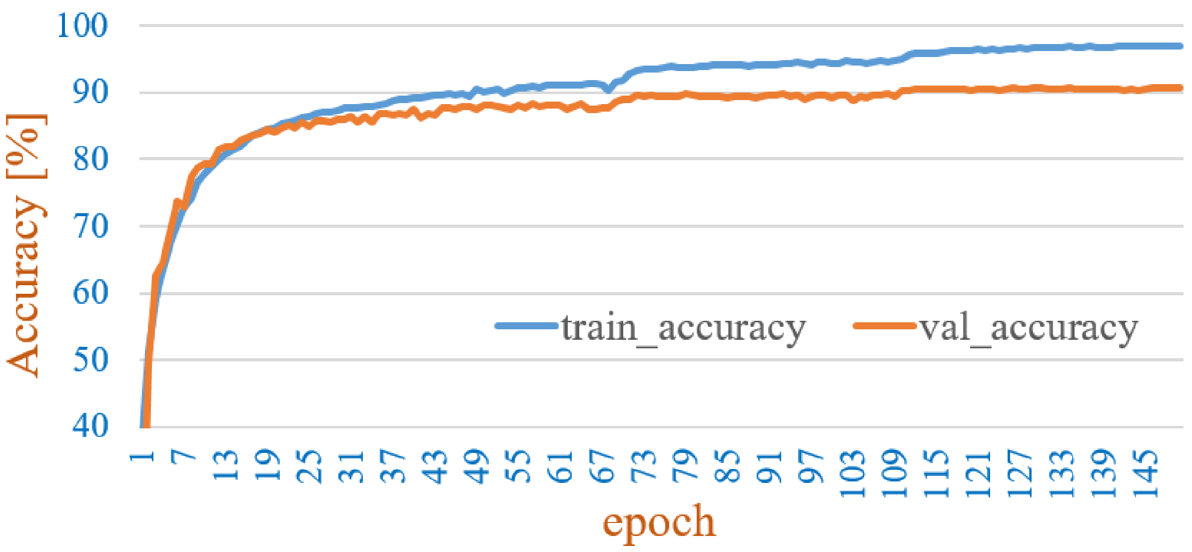




Model Generalization

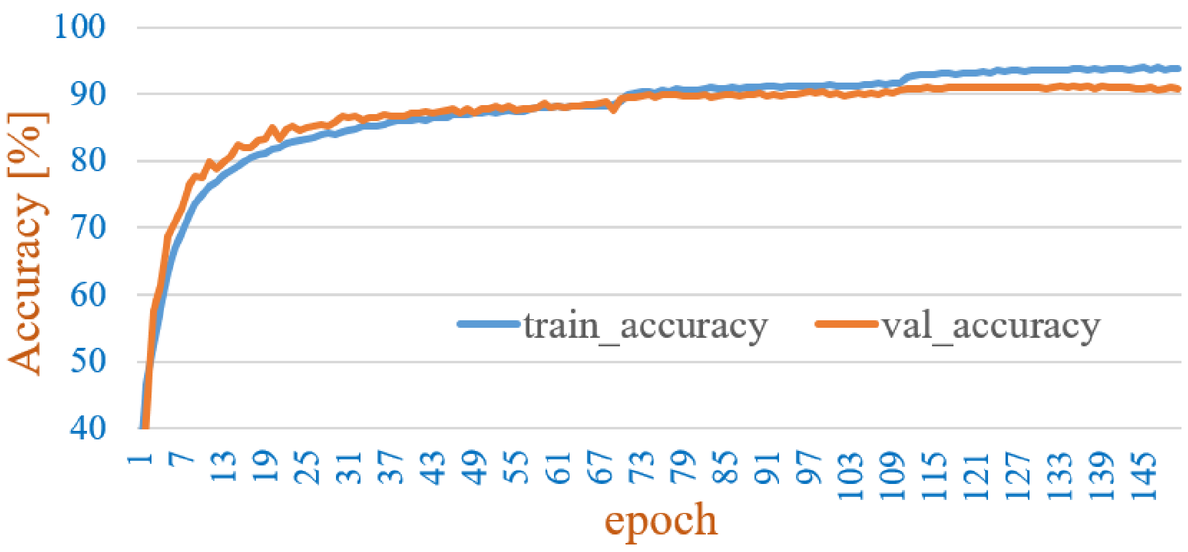
❖ Trick 5: Data augmentation

Horizontal flip



val_accuracy reaches to ~90.7%

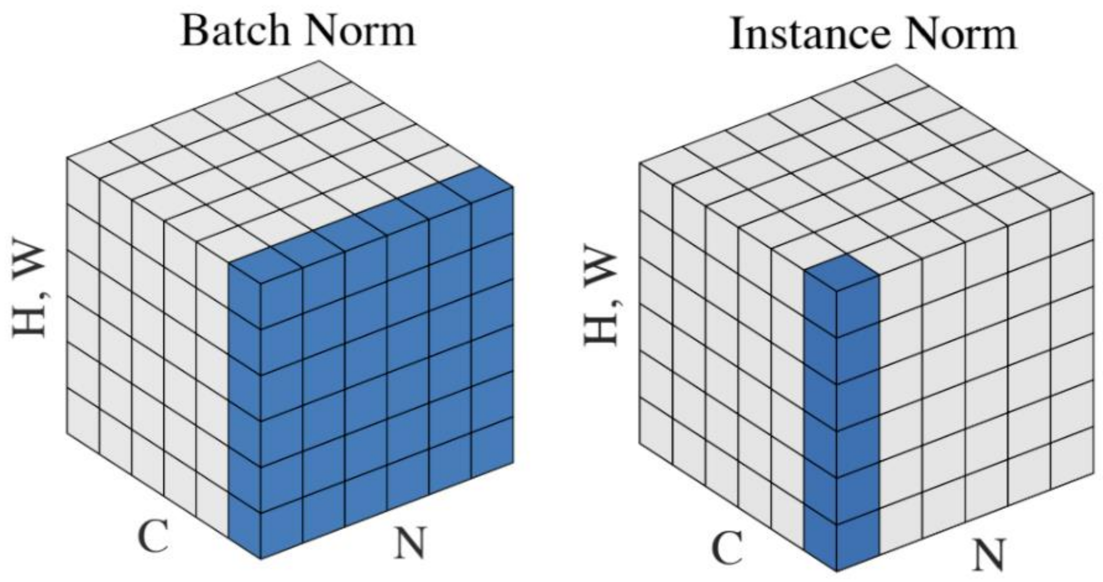
Horizontal flip + crop-and-resize



val_accuracy reaches to ~91.2%

Model Generalization

❖ Trick 6: Instance normalization



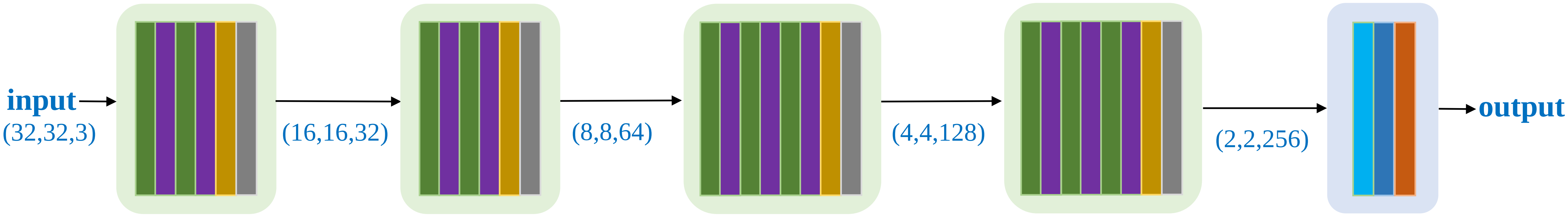
https://arxiv.org/pdf/1803.08494.pdf

"applying IN which does not only reduce the difference caused by domain changes, but also the illumination variation in single spectral images"

AFD-Net Aggregated Feature Difference Learning for Cross-Spectral Image Patch Matching (ICCV, 2019)

Model Generalization

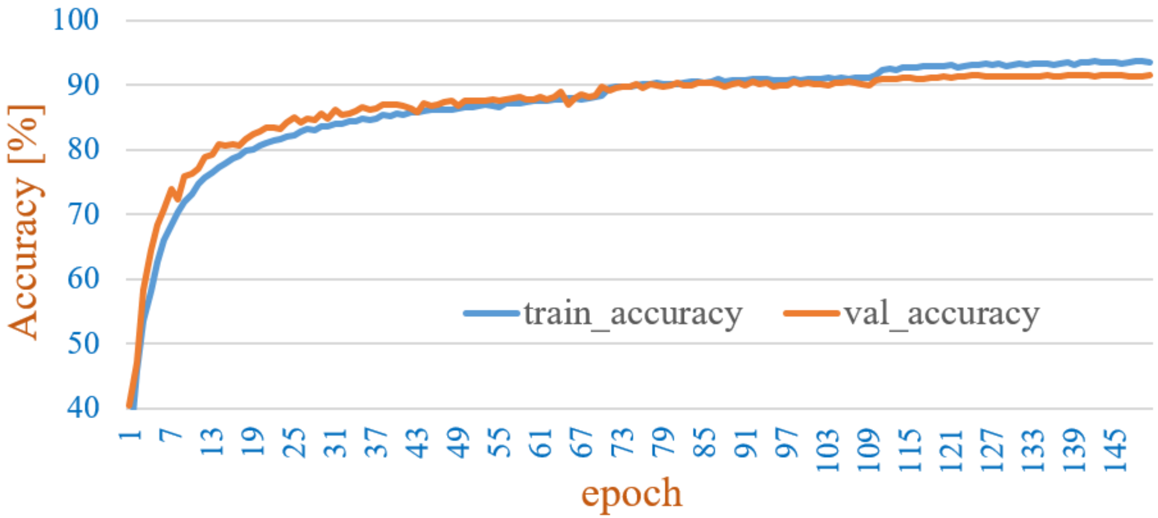
❖ Trick 6: Instance normalization



Dropout

Instance Normalization +
Batch normalization

(3x3) Convolution
padding='same', stride=1 +
ReLU + kernel regularization

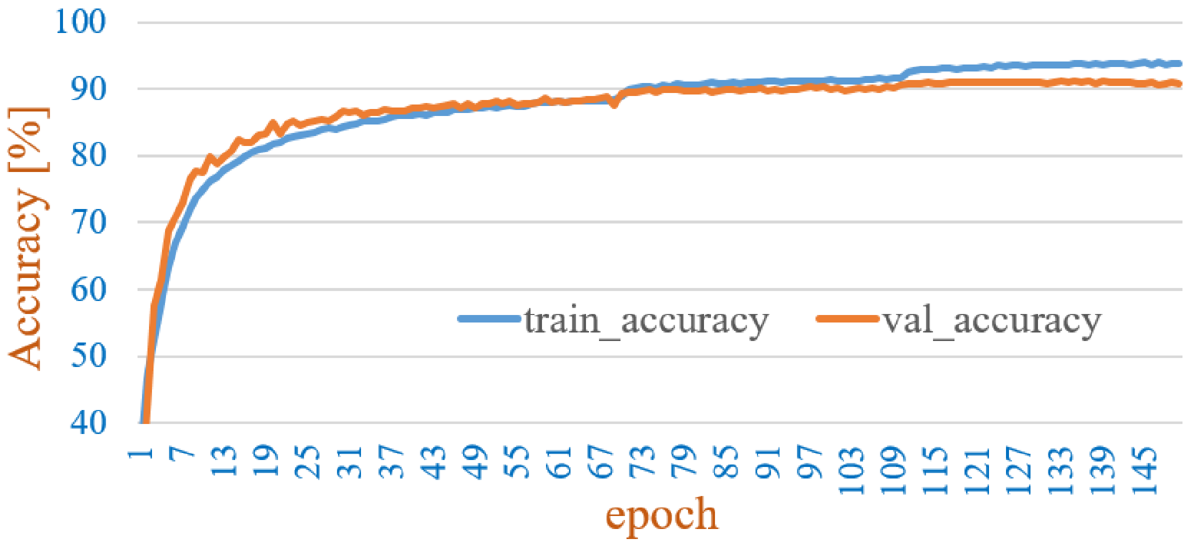


val_accuracy reaches to ~91.6%

Model Generalization

❖ Summary

Horizontal flip + crop-and-resize



val_accuracy reaches to ~91.6%

train_accuracy reaches to ~93.7%

Batch normalization
Dropout
Kernel regularization
Data augmentation

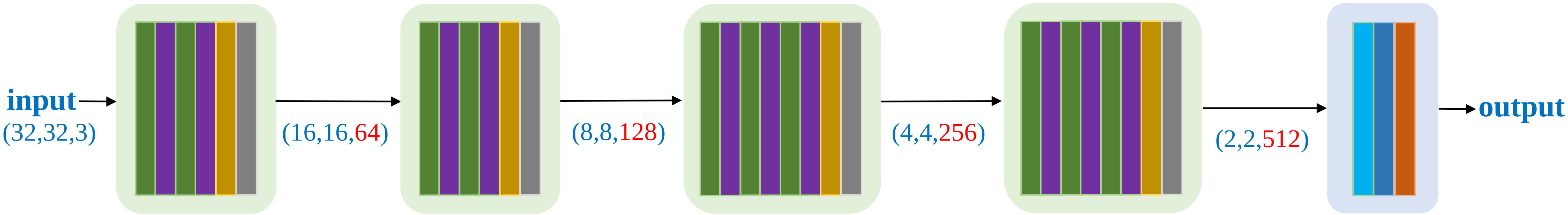
Idea: try to increase train_accuracy, expect val_accuracy increases too

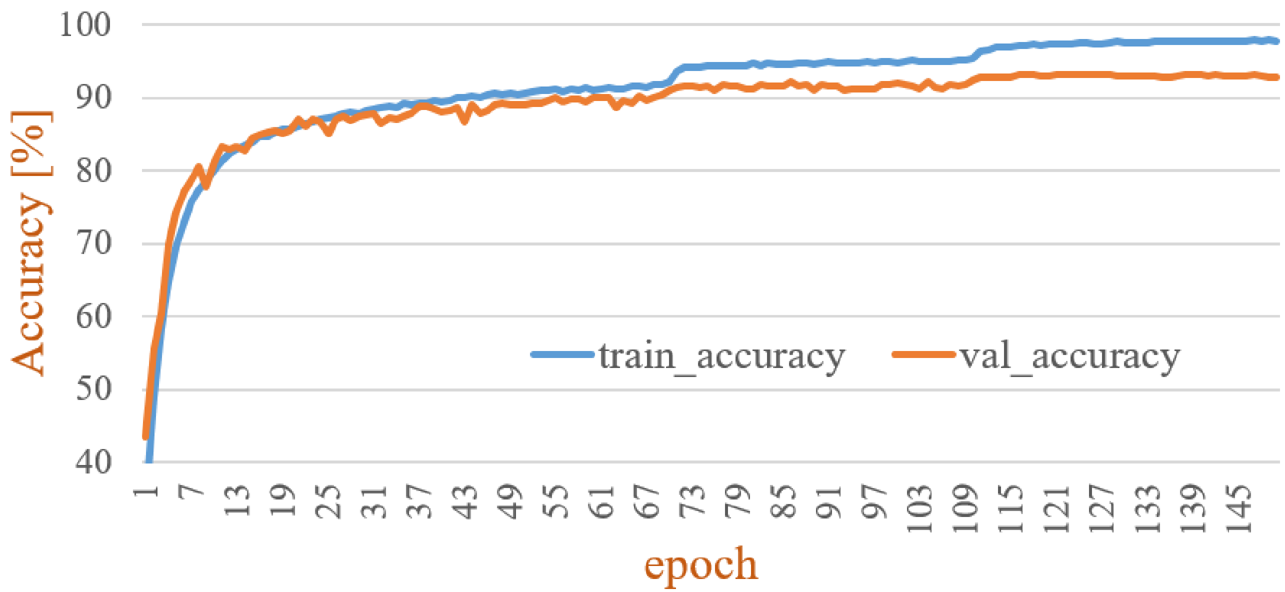
➡ **Increase model capacity**

AI VIETNAM
Free AI Course

Model Generalization

❖ Increase model capacity





val_accuracy reaches to ~93.6%

train_accuracy reaches to ~97.9%

Model Generalization

- **Is**

Summary

Thank you!