



E-SAN THAILAND CODING & AI ACADEMY

โครงการวิจัยโมเดลระบบนิเวศการเรียนรู้ที่บูรณาการ CODING & AI สำหรับเยาวชน
Model of Learning Ecosystem Platform integrate with Coding & AI for Youth



โครงการย่อยที่ 6

การพัฒนาเยาวชนเพื่อเข้าสู่วิชาชีพขั้นสูงด้าน Coding & AI
ร่วมกับ Coding Entrepreneur & Partnership: **Personal AI**

BiTNet: AI for Ultrasound Image Classification

ผศ.ดร.ธนพงศ์ อินทะ
ผู้เชี่ยวชาญด้าน Computer Vision

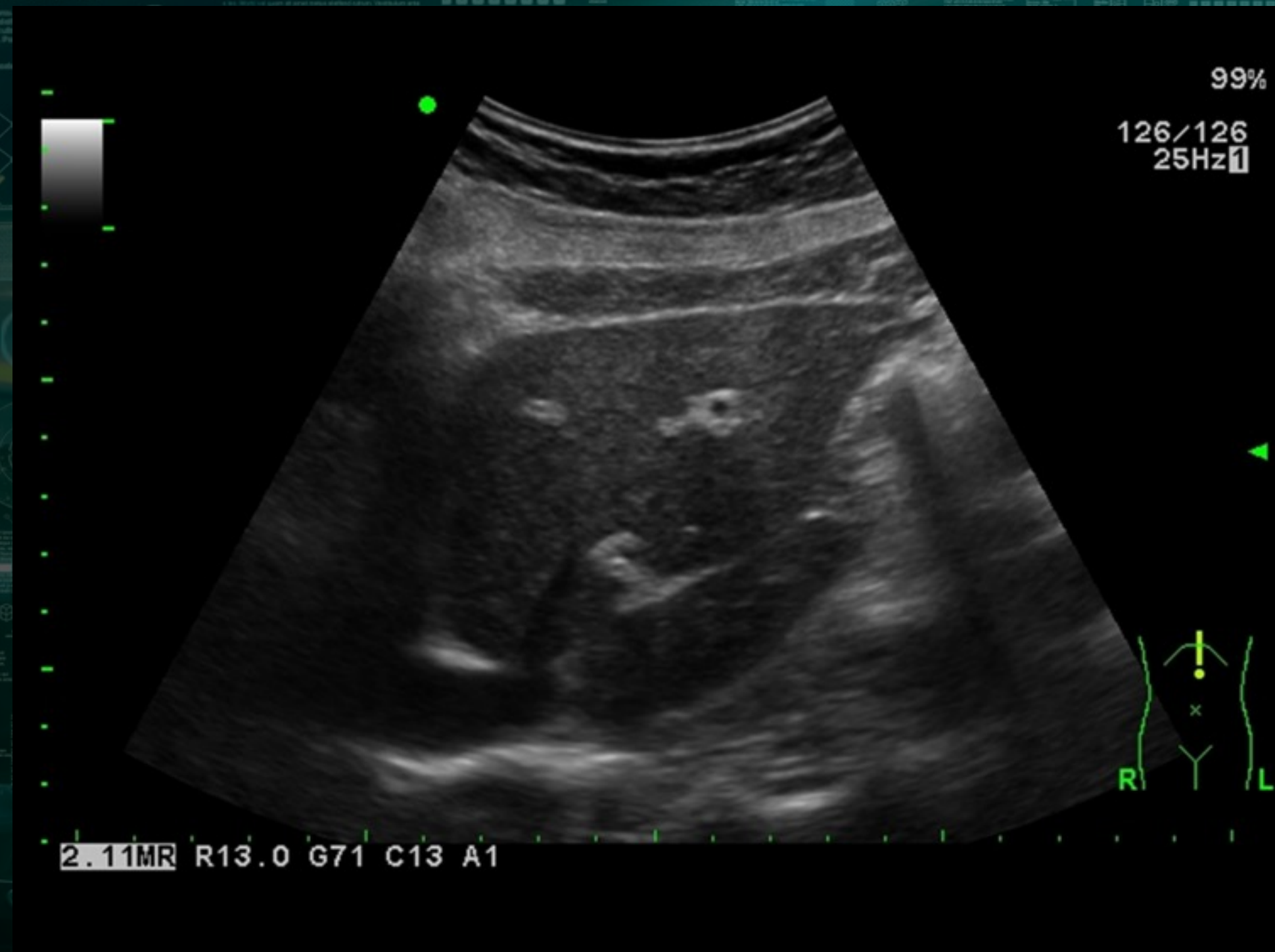
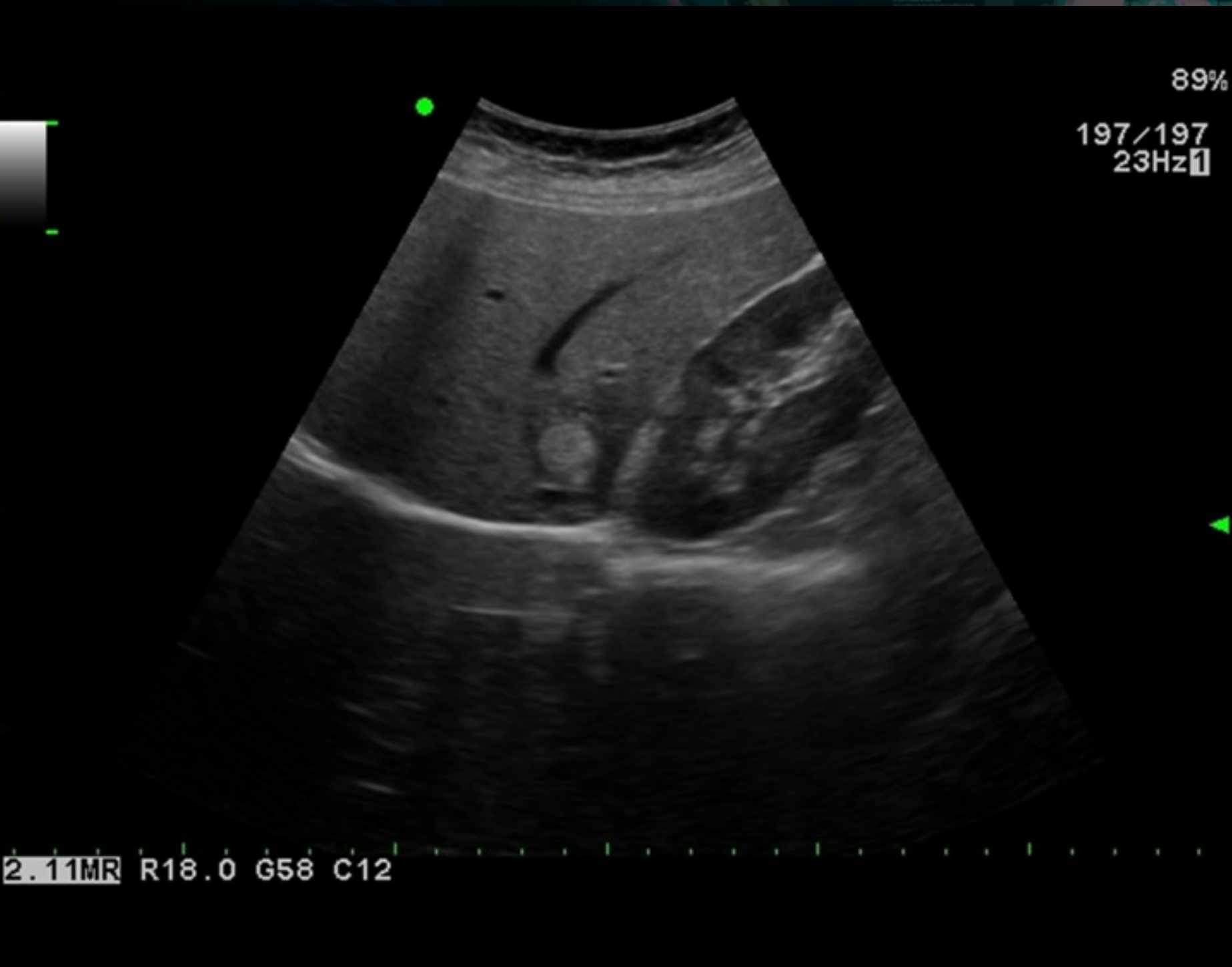


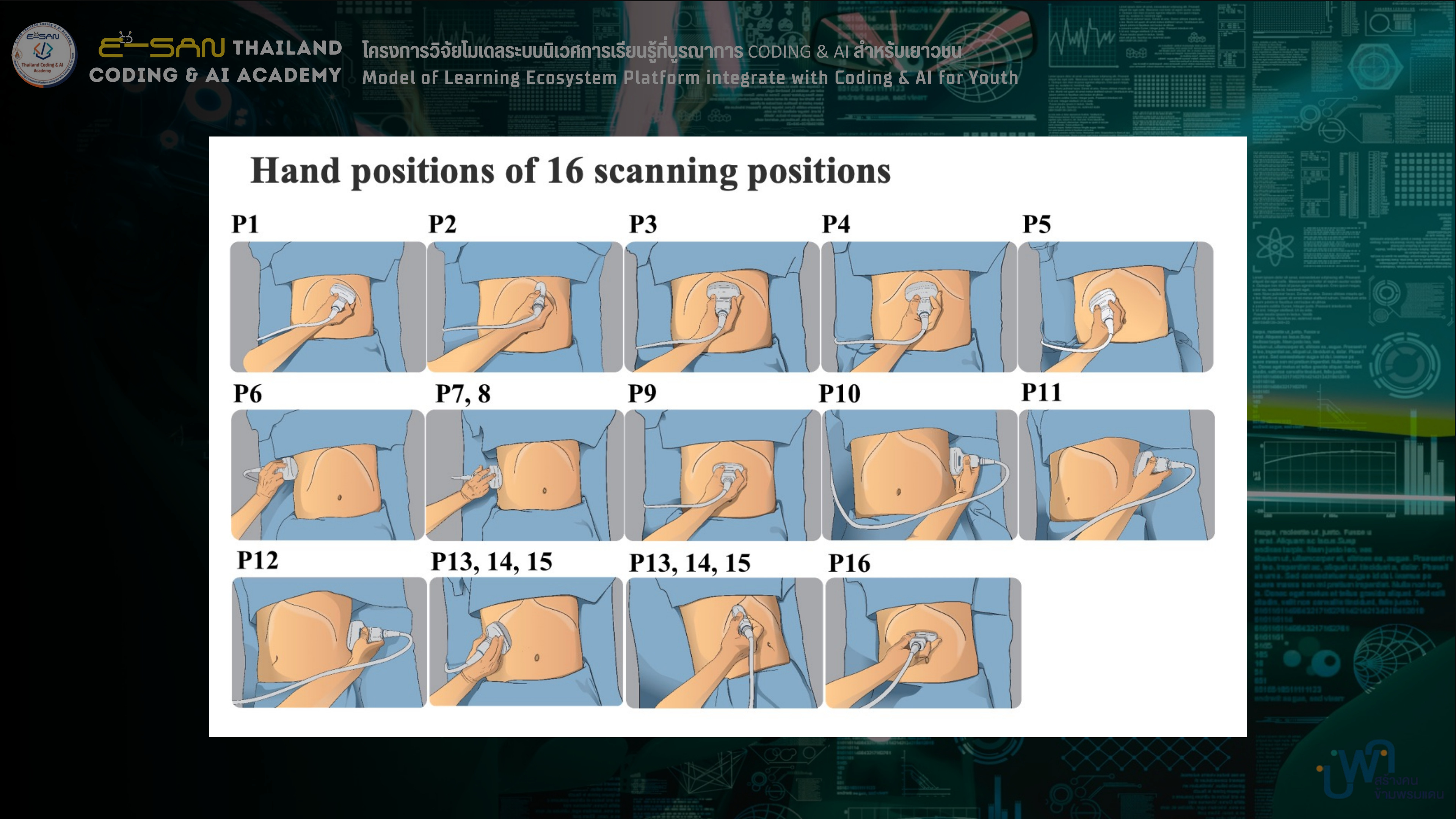
E-SAN THAILAND
CODING & AI ACADEMY

โครงการวิจัยโมเดลระบบนิเวศการเรียนรู้ที่บูรณาการ CODING & AI สำหรับเยาวชน
Model of Learning Ecosystem Platform integrate with Coding & AI for Youth

Add a little bit of body text

Data preparation



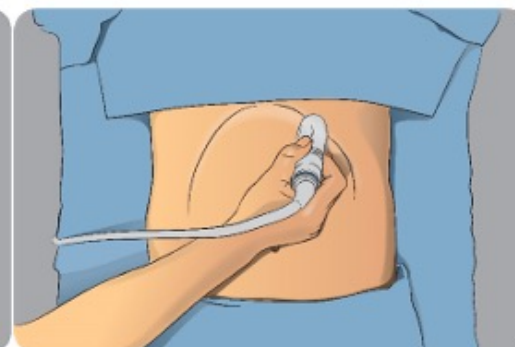


Hand positions of 16 scanning positions

P1



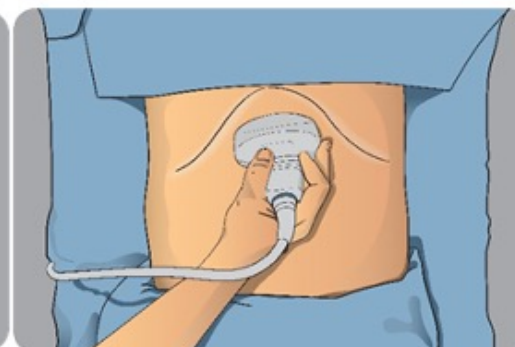
P2



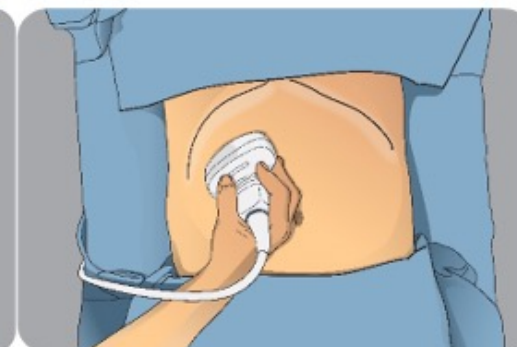
P3



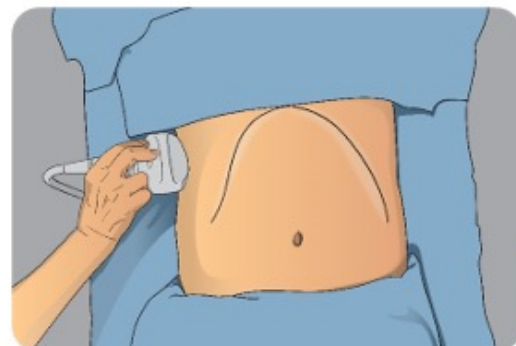
P4



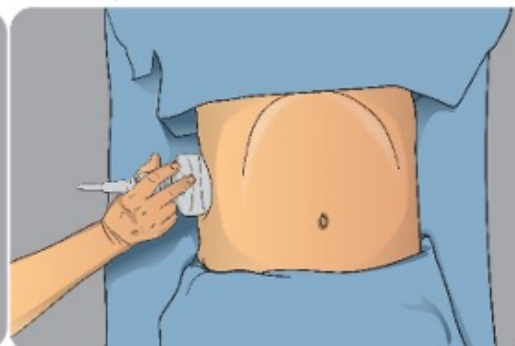
P5



P6



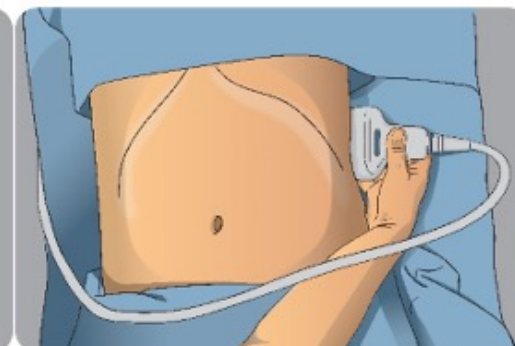
P7, 8



P9



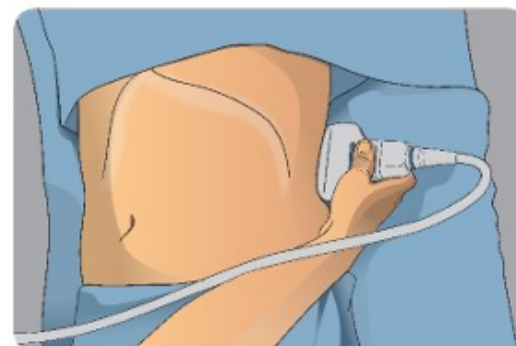
P10



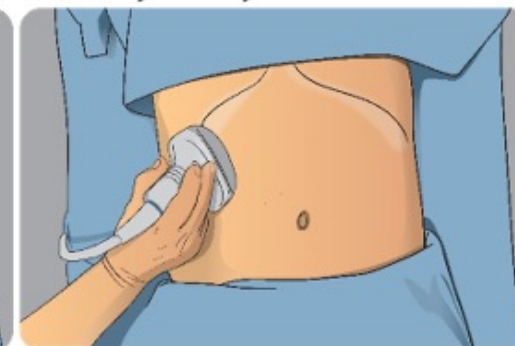
P11



P12



P13, 14, 15

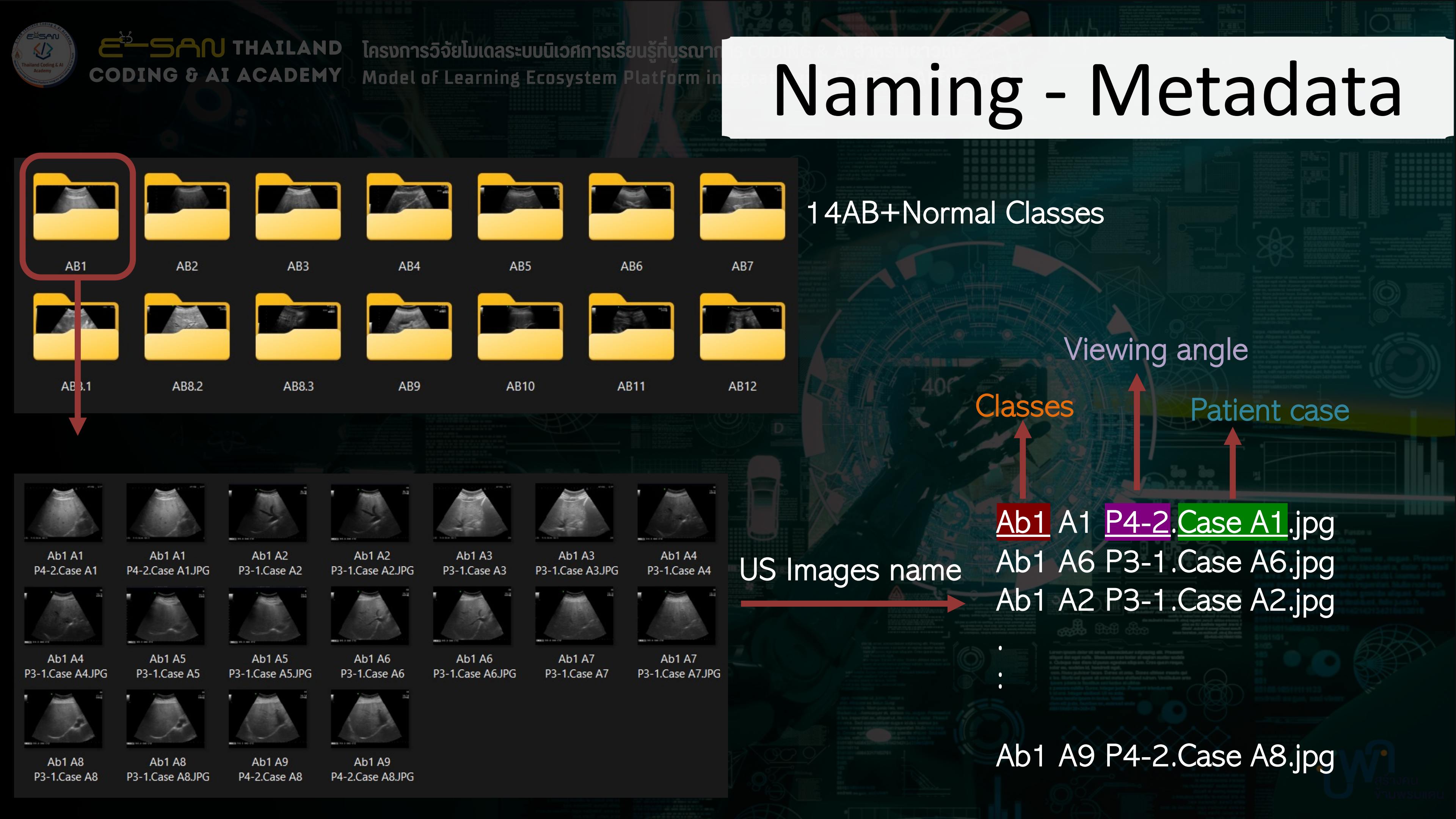


P13, 14, 15

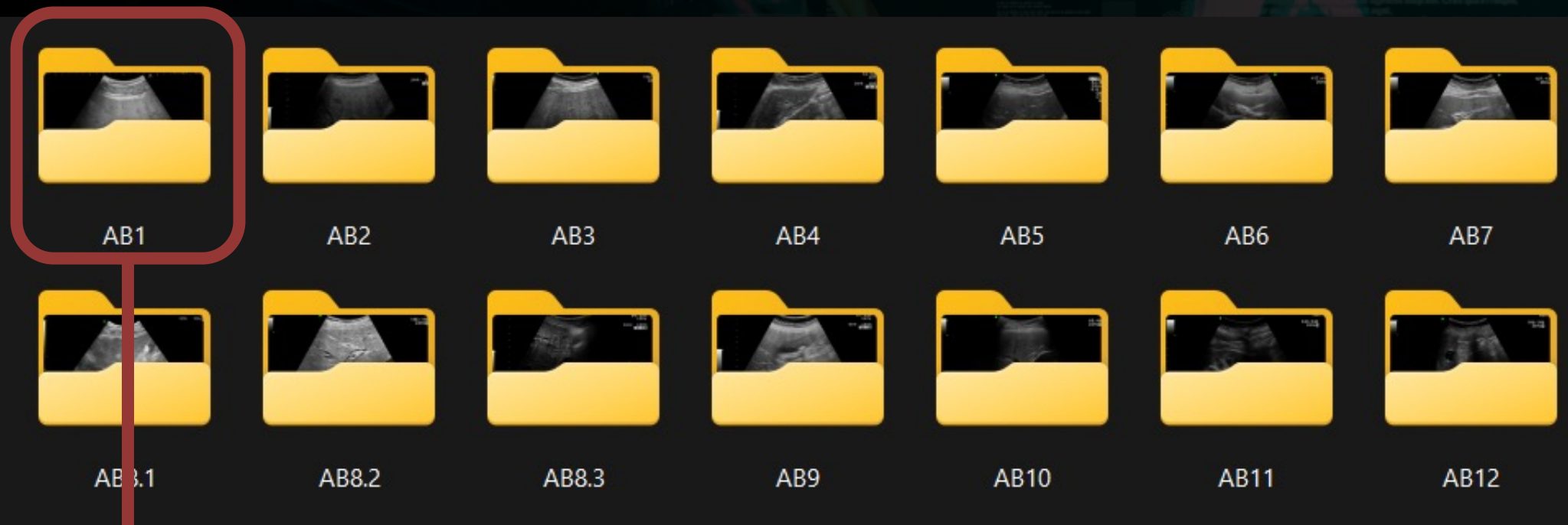


P16

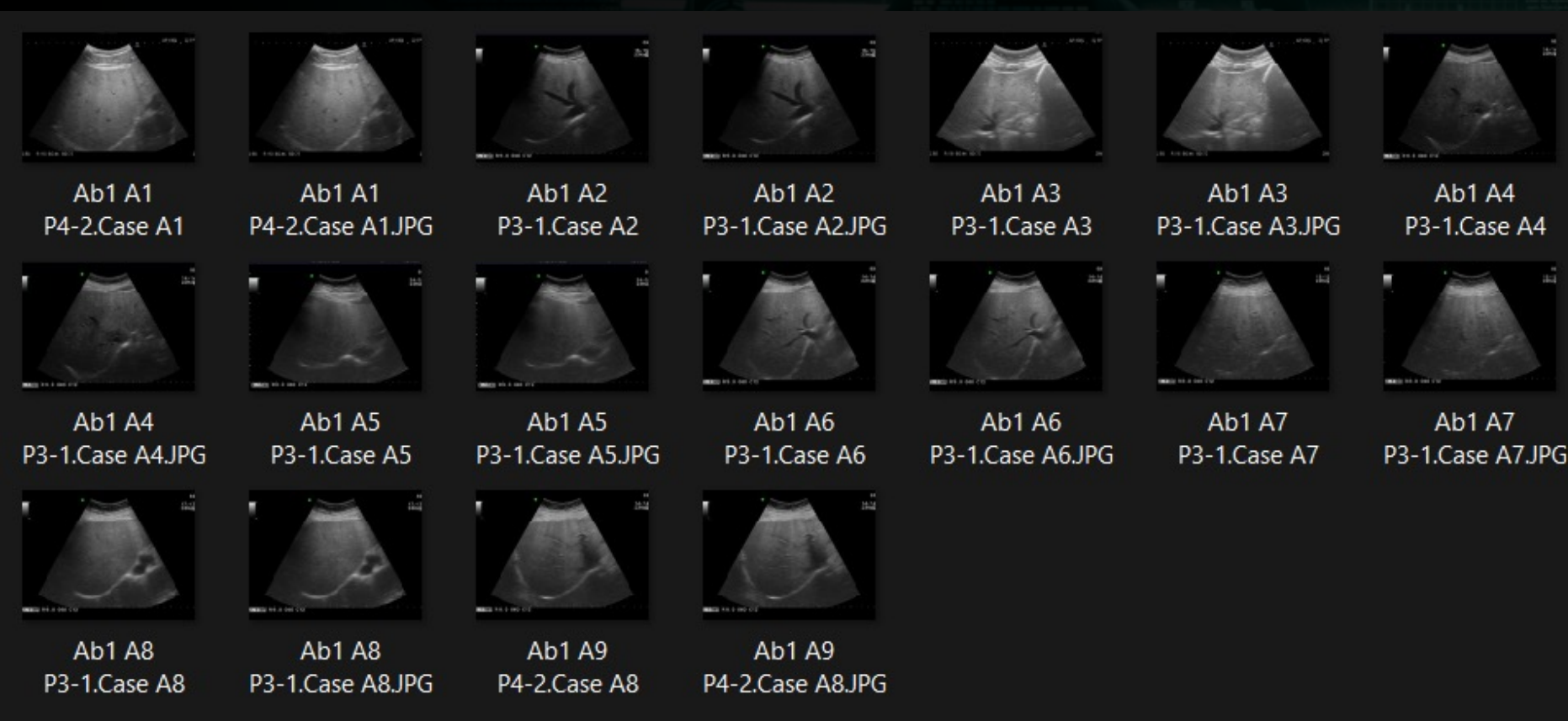




Naming - Metadata



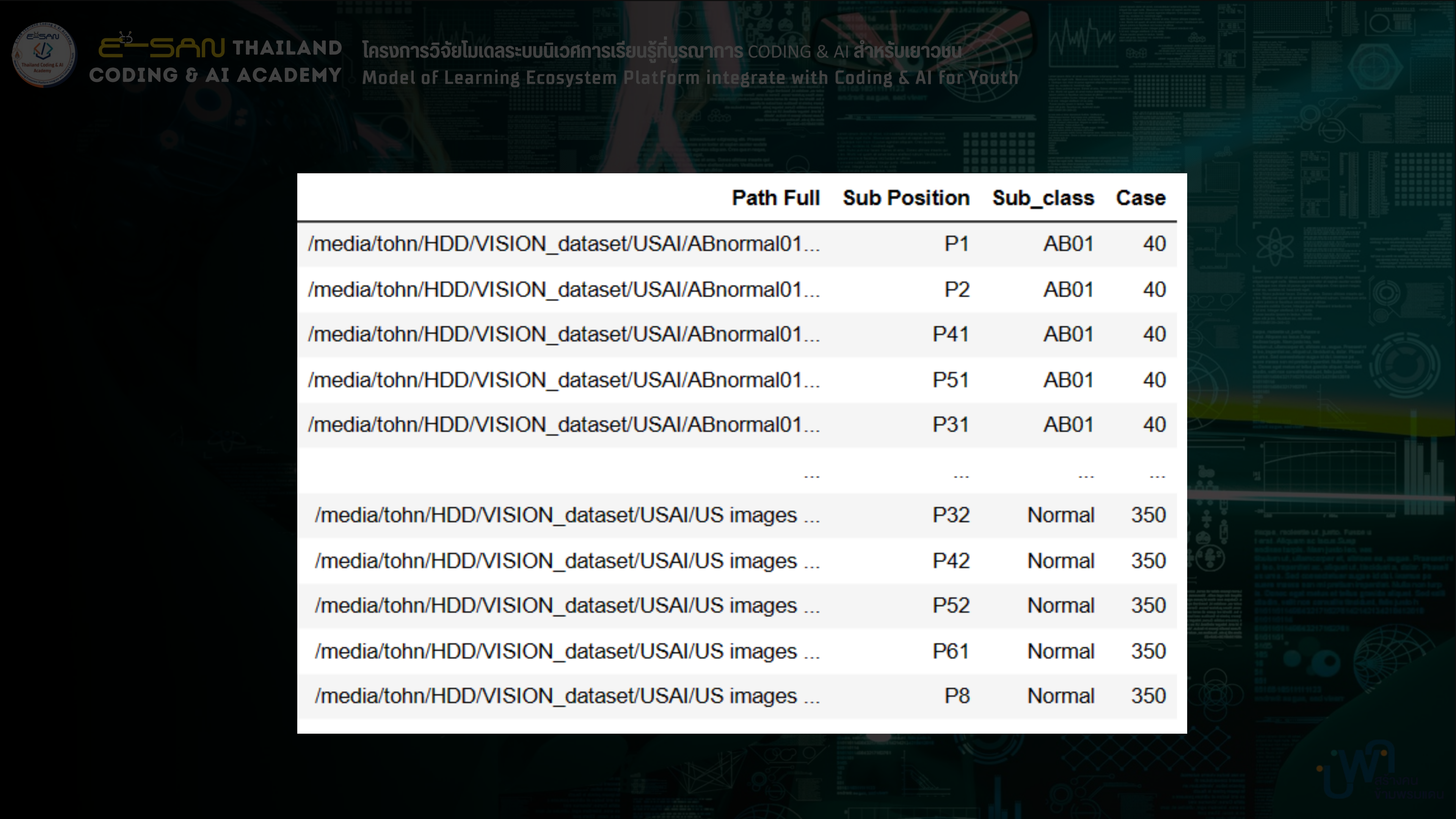
14AB+Normal Classes



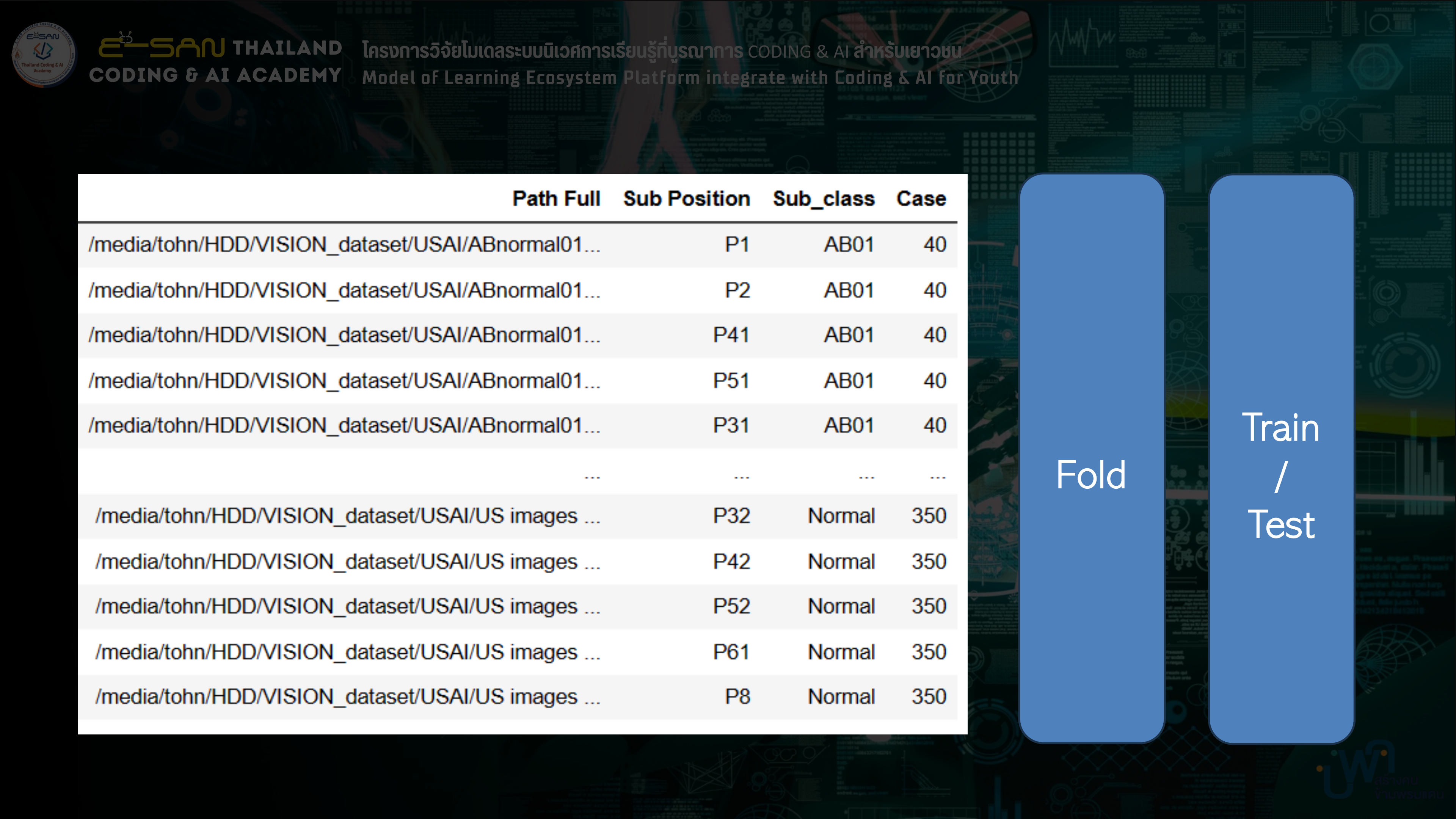
US Images name

Classes
Viewing angle
Patient case

Ab1 A1 P4-2.Case A1.jpg
Ab1 A6 P3-1.Case A6.jpg
Ab1 A2 P3-1.Case A2.jpg
:
Ab1 A9 P4-2.Case A8.jpg



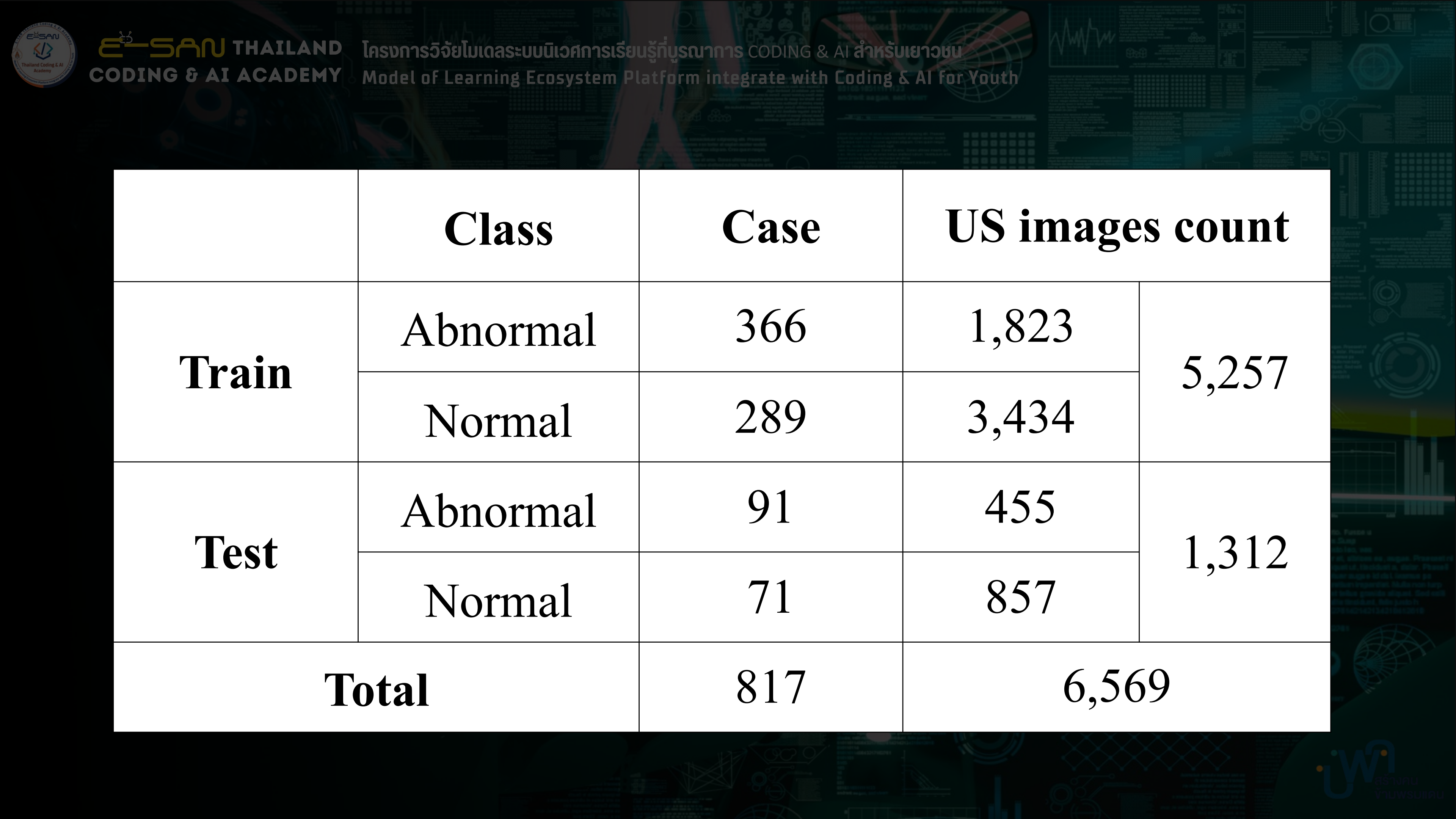
Path Full	Sub Position	Sub_class	Case
/media/tohn/HDD/VISION_dataset/USAI/ABnormal01...	P1	AB01	40
/media/tohn/HDD/VISION_dataset/USAI/ABnormal01...	P2	AB01	40
/media/tohn/HDD/VISION_dataset/USAI/ABnormal01...	P41	AB01	40
/media/tohn/HDD/VISION_dataset/USAI/ABnormal01...	P51	AB01	40
/media/tohn/HDD/VISION_dataset/USAI/ABnormal01...	P31	AB01	40
...
/media/tohn/HDD/VISION_dataset/USAI/US images ...	P32	Normal	350
/media/tohn/HDD/VISION_dataset/USAI/US images ...	P42	Normal	350
/media/tohn/HDD/VISION_dataset/USAI/US images ...	P52	Normal	350
/media/tohn/HDD/VISION_dataset/USAI/US images ...	P61	Normal	350
/media/tohn/HDD/VISION_dataset/USAI/US images ...	P8	Normal	350



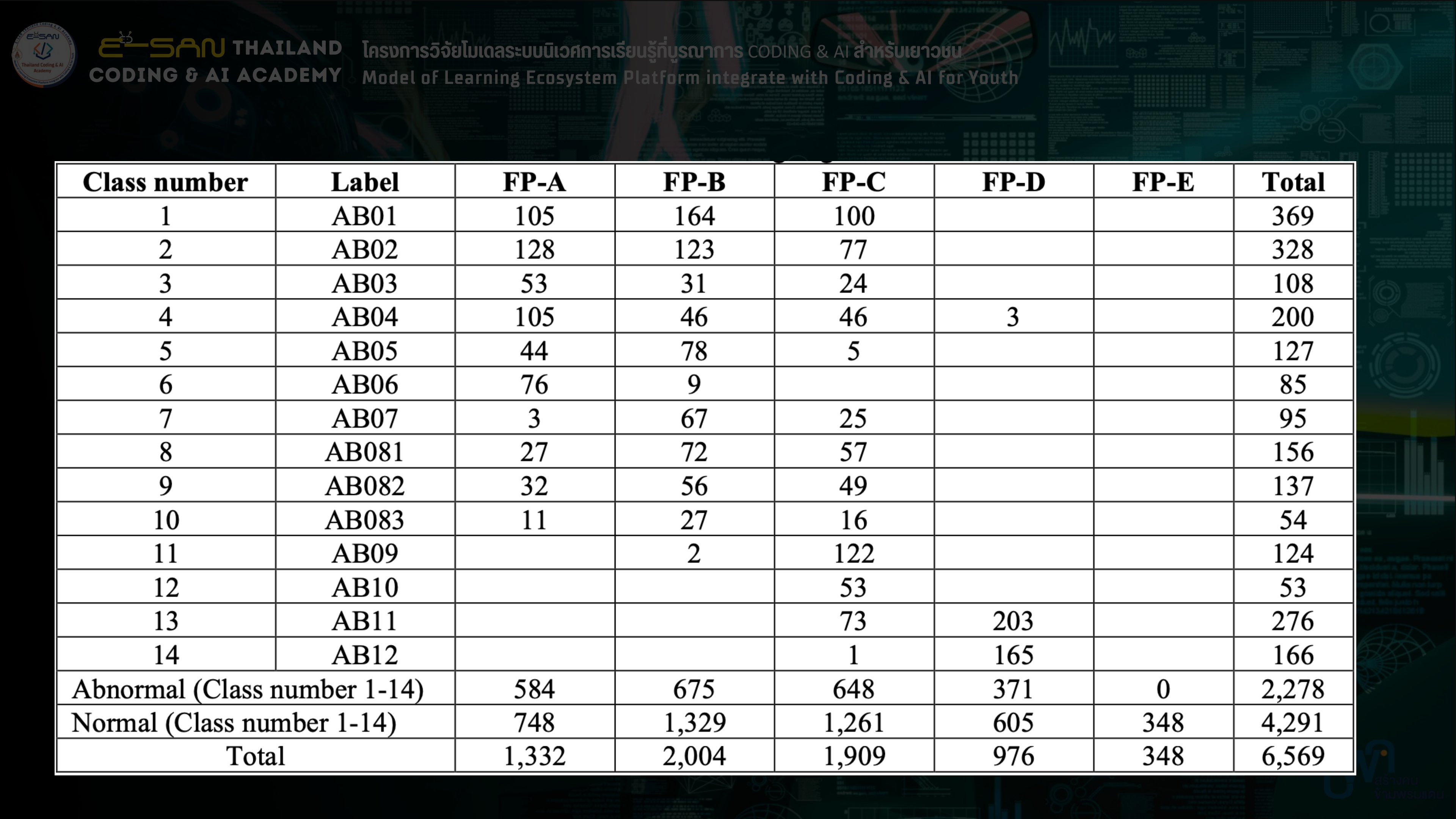
Path Full	Sub Position	Sub_class	Case
/media/tohn/HDD/VISION_dataset/USAI/ABnormal01...	P1	AB01	40
/media/tohn/HDD/VISION_dataset/USAI/ABnormal01...	P2	AB01	40
/media/tohn/HDD/VISION_dataset/USAI/ABnormal01...	P41	AB01	40
/media/tohn/HDD/VISION_dataset/USAI/ABnormal01...	P51	AB01	40
/media/tohn/HDD/VISION_dataset/USAI/ABnormal01...	P31	AB01	40
...
/media/tohn/HDD/VISION_dataset/USAI/US images ...	P32	Normal	350
/media/tohn/HDD/VISION_dataset/USAI/US images ...	P42	Normal	350
/media/tohn/HDD/VISION_dataset/USAI/US images ...	P52	Normal	350
/media/tohn/HDD/VISION_dataset/USAI/US images ...	P61	Normal	350
/media/tohn/HDD/VISION_dataset/USAI/US images ...	P8	Normal	350

Fold

Train
/
Test



	Class	Case	US images count	
Train	Abnormal	366	1,823	5,257
	Normal	289	3,434	
Test	Abnormal	91	455	1,312
	Normal	71	857	
Total		817	6,569	

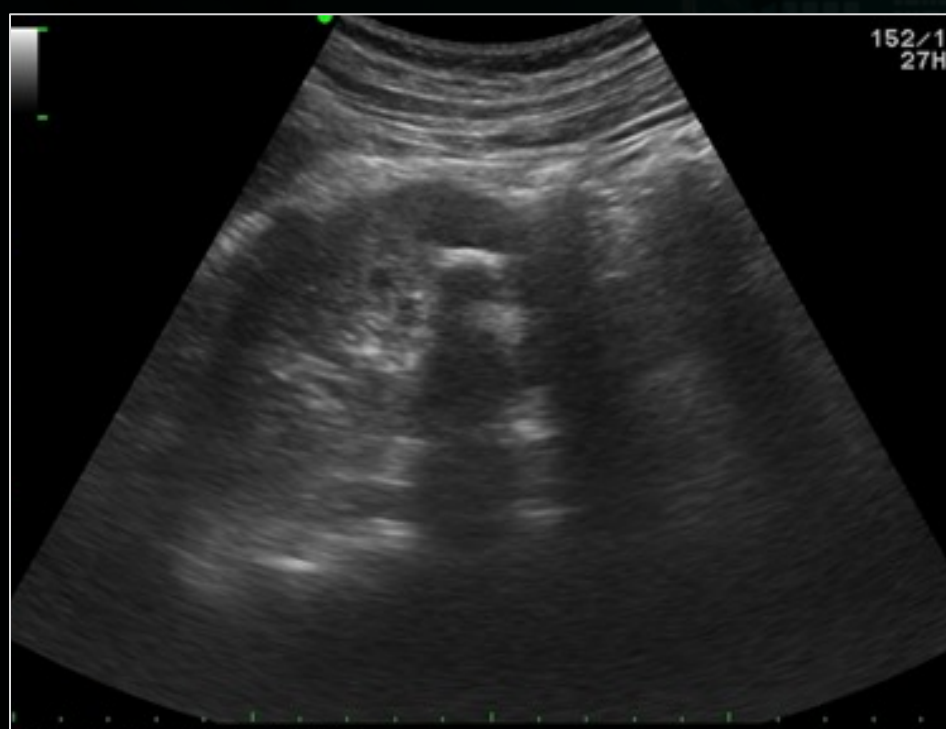


Class number	Label	FP-A	FP-B	FP-C	FP-D	FP-E	Total
1	AB01	105	164	100			369
2	AB02	128	123	77			328
3	AB03	53	31	24			108
4	AB04	105	46	46	3		200
5	AB05	44	78	5			127
6	AB06	76	9				85
7	AB07	3	67	25			95
8	AB081	27	72	57			156
9	AB082	32	56	49			137
10	AB083	11	27	16			54
11	AB09		2	122			124
12	AB10			53			53
13	AB11			73	203		276
14	AB12			1	165		166
Abnormal (Class number 1-14)		584	675	648	371	0	2,278
Normal (Class number 1-14)		748	1,329	1,261	605	348	4,291
Total		1,332	2,004	1,909	976	348	6,569

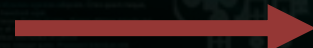


Input Size

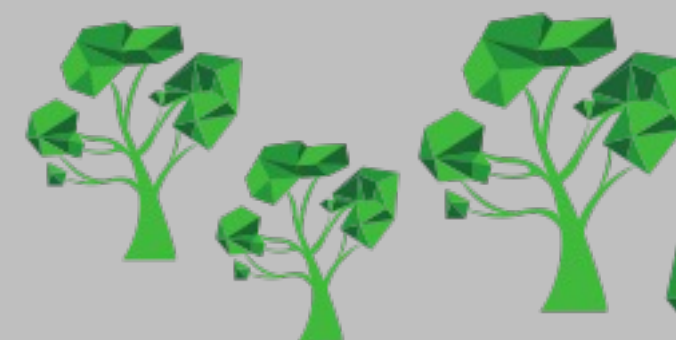
BiTNet

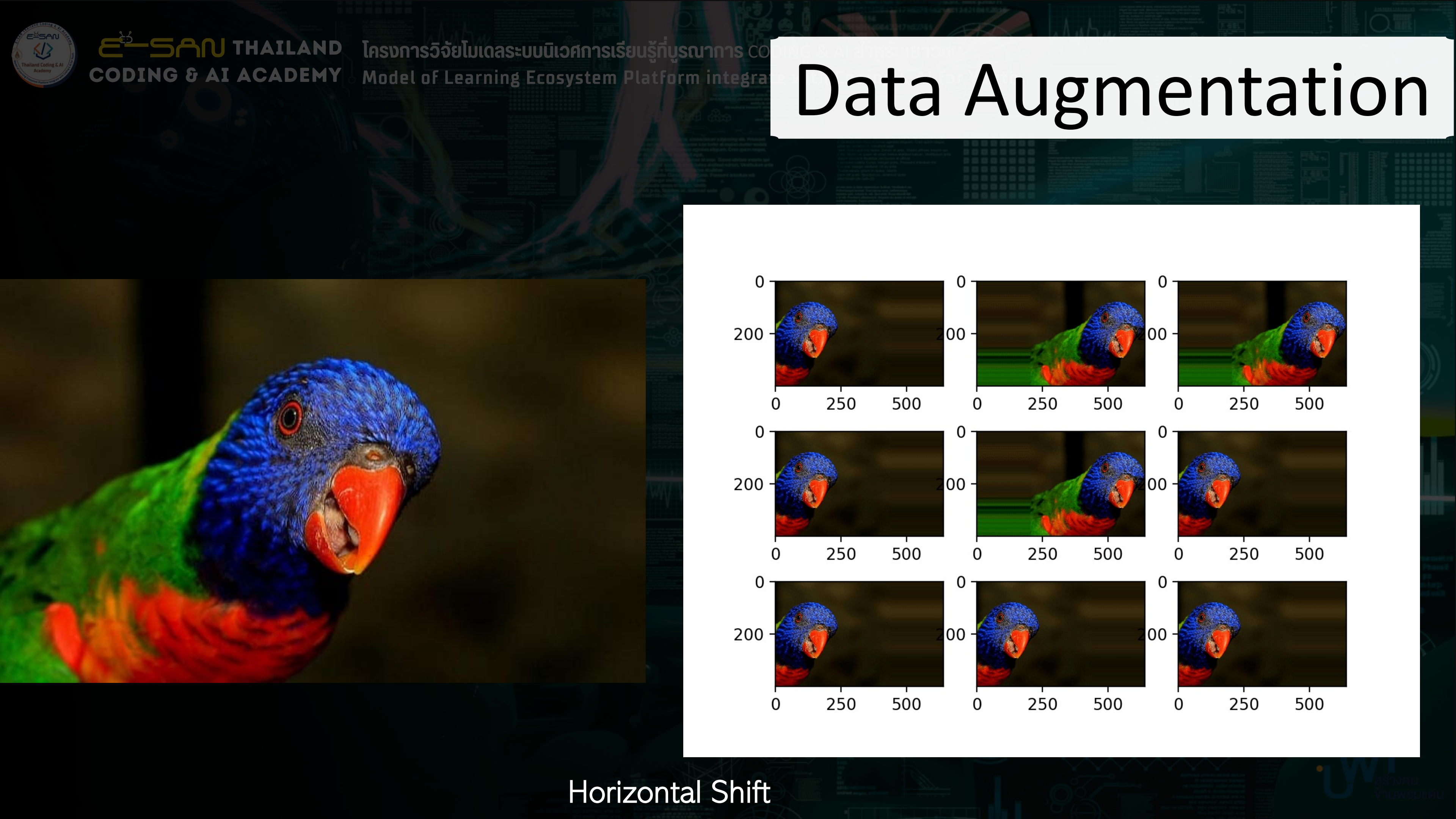


456x456x3

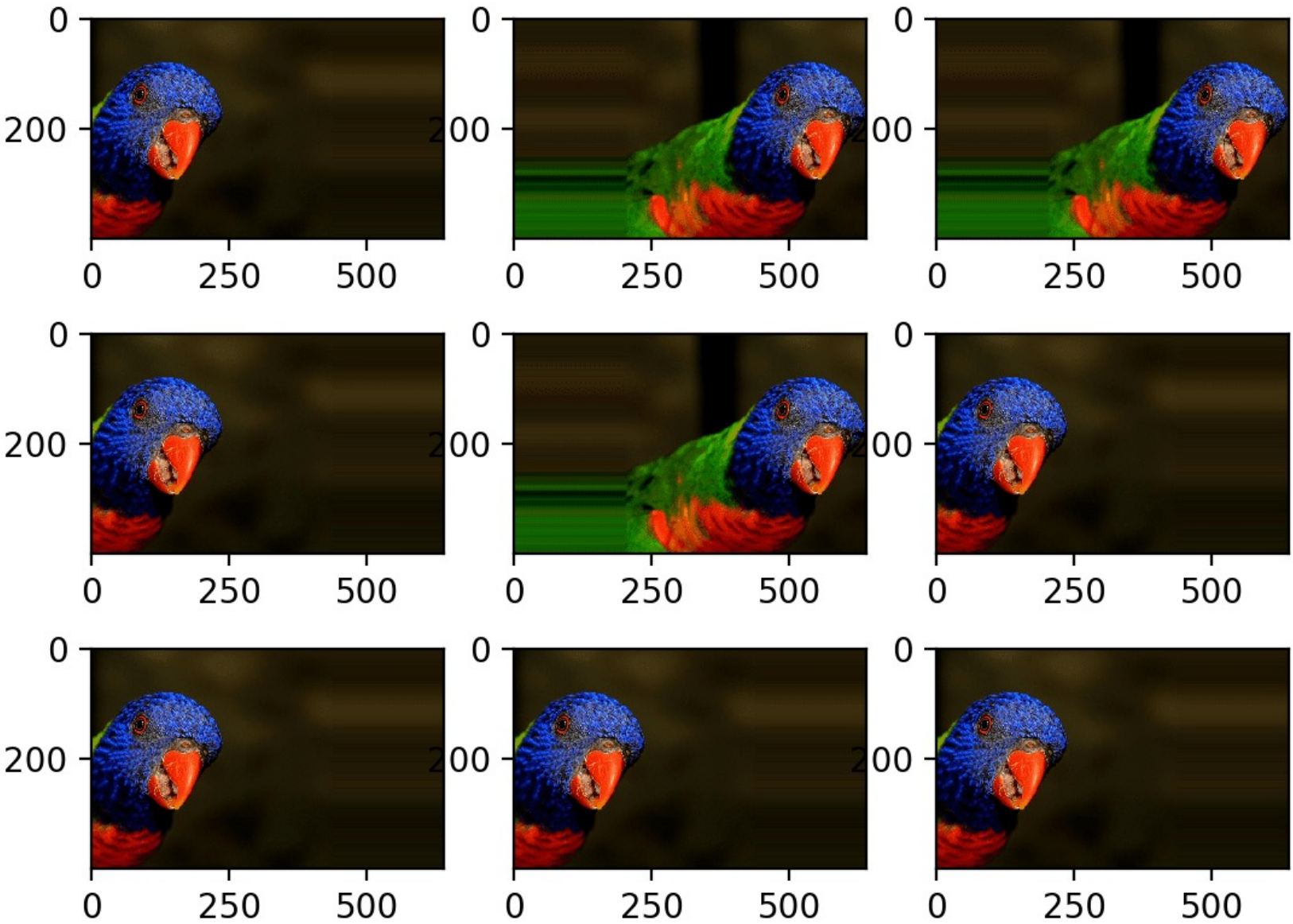


EfficientNetB5

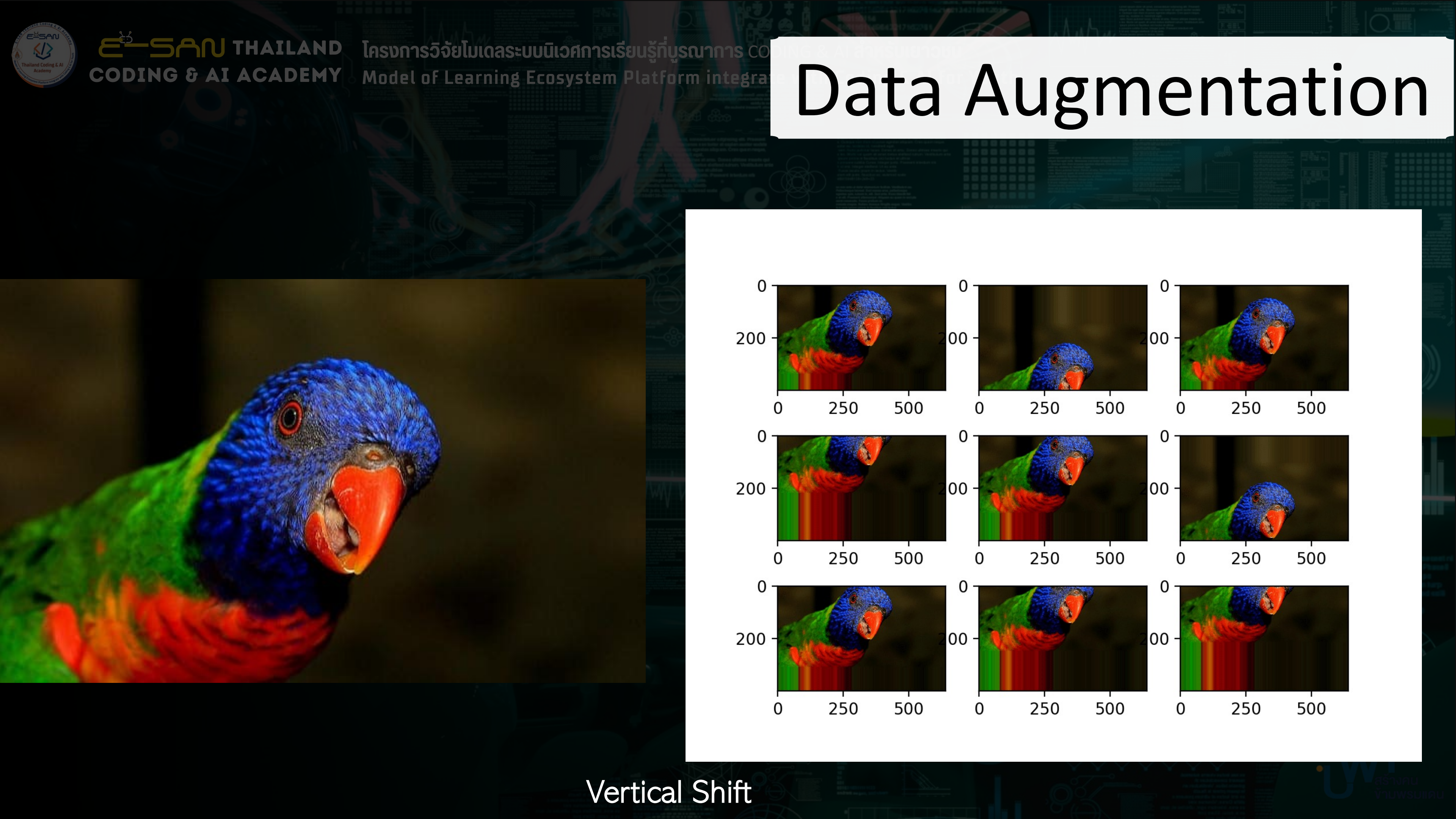




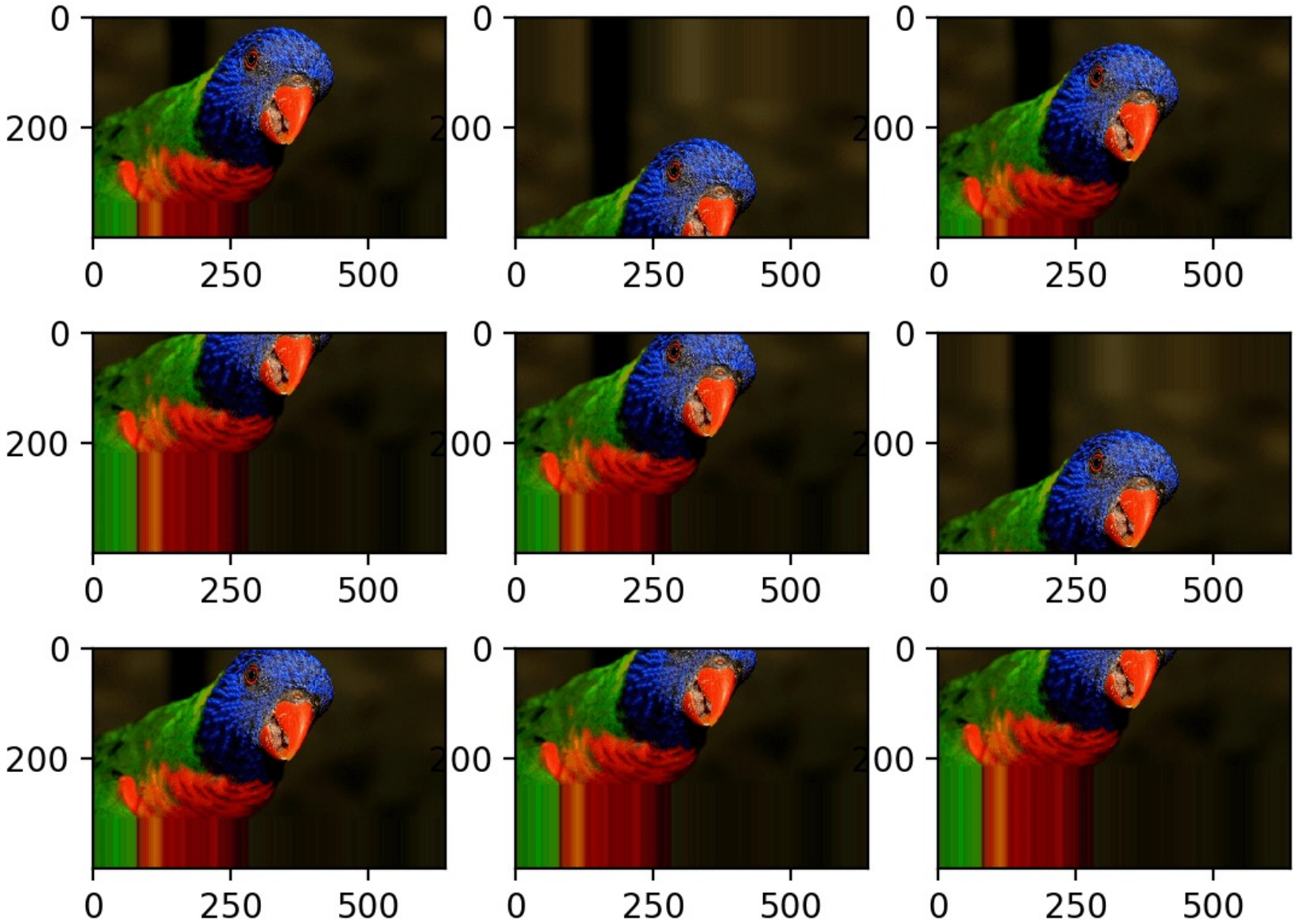
Data Augmentation



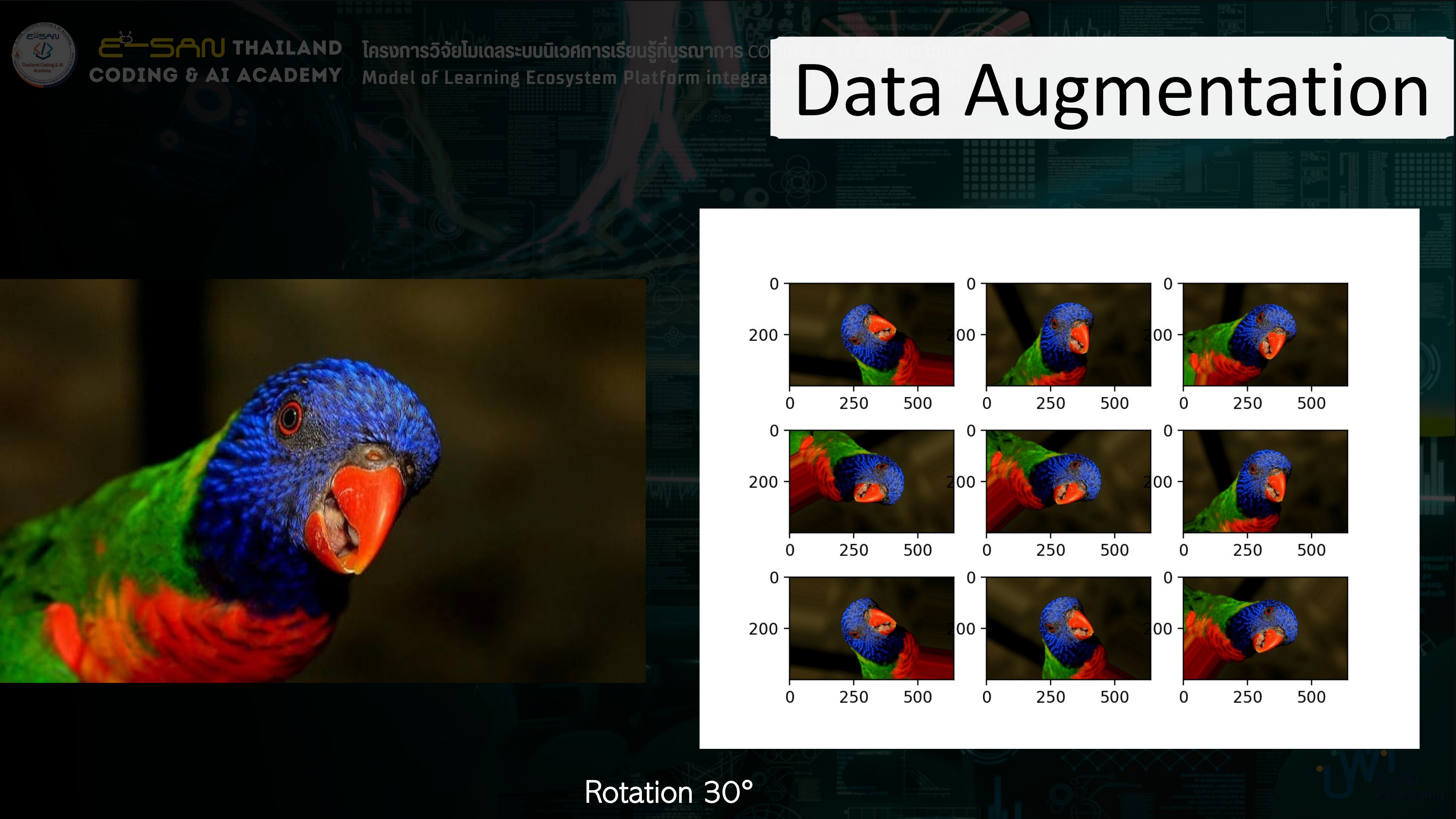
Horizontal Shift



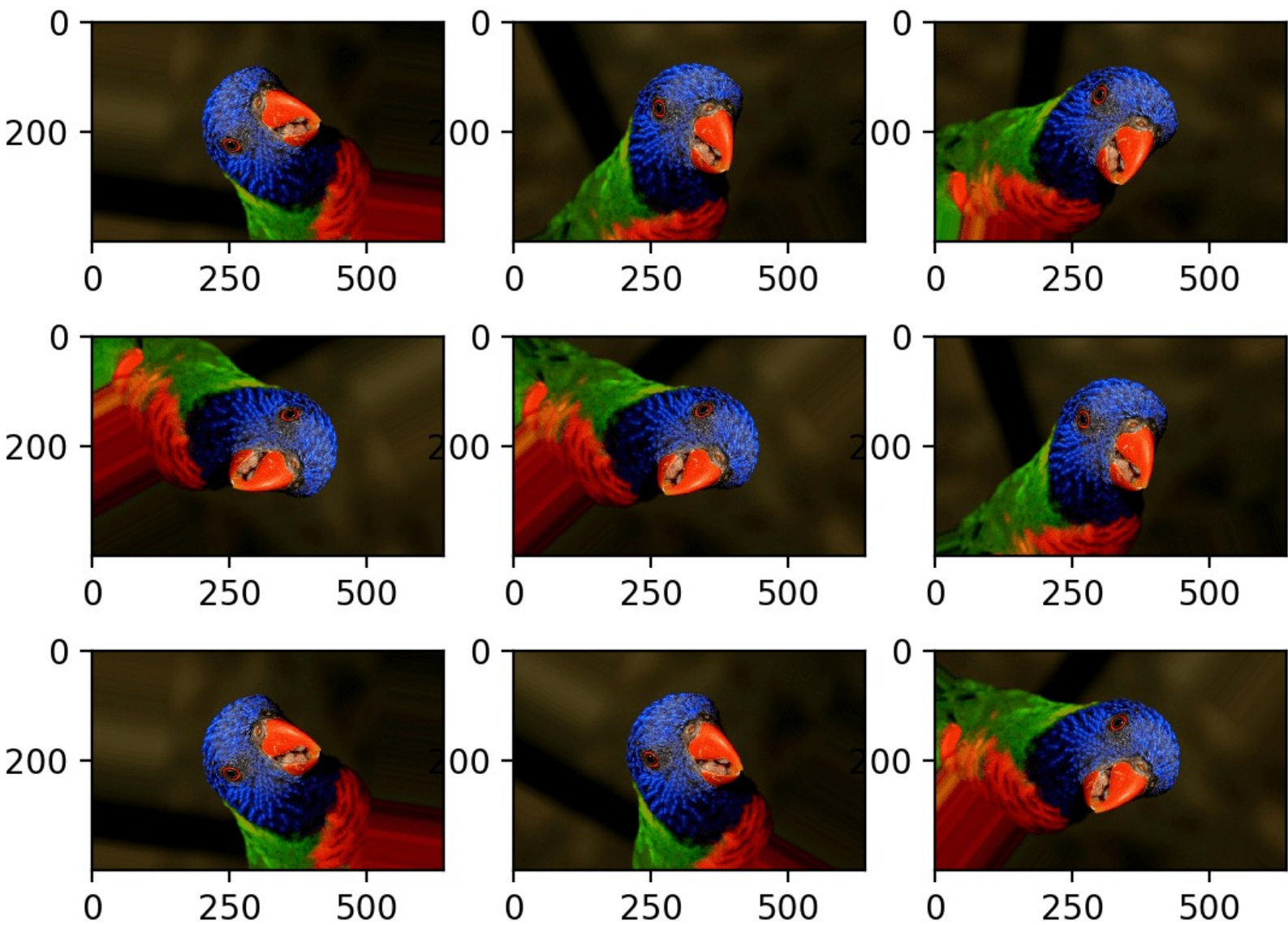
Data Augmentation



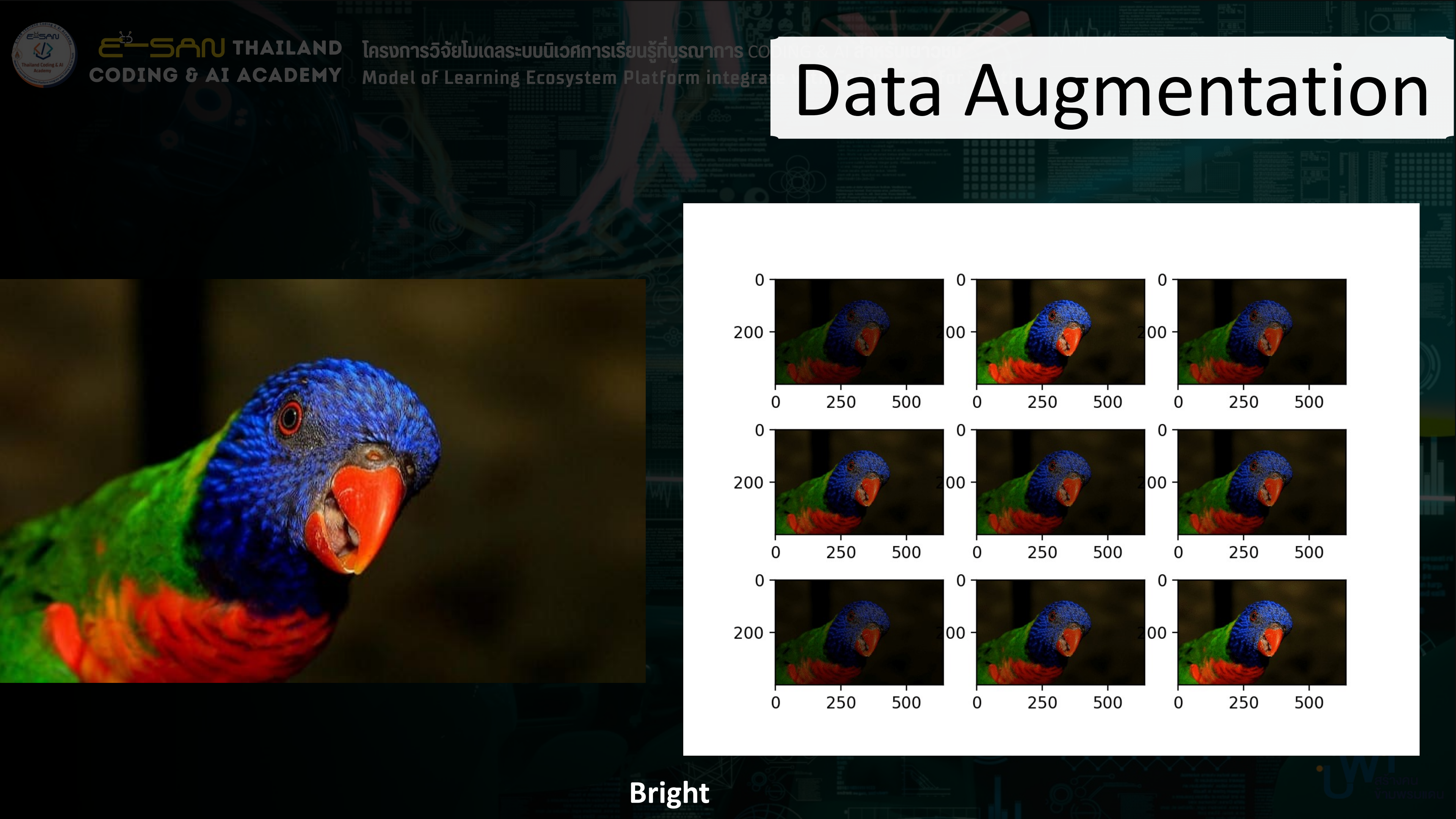
Vertical Shift



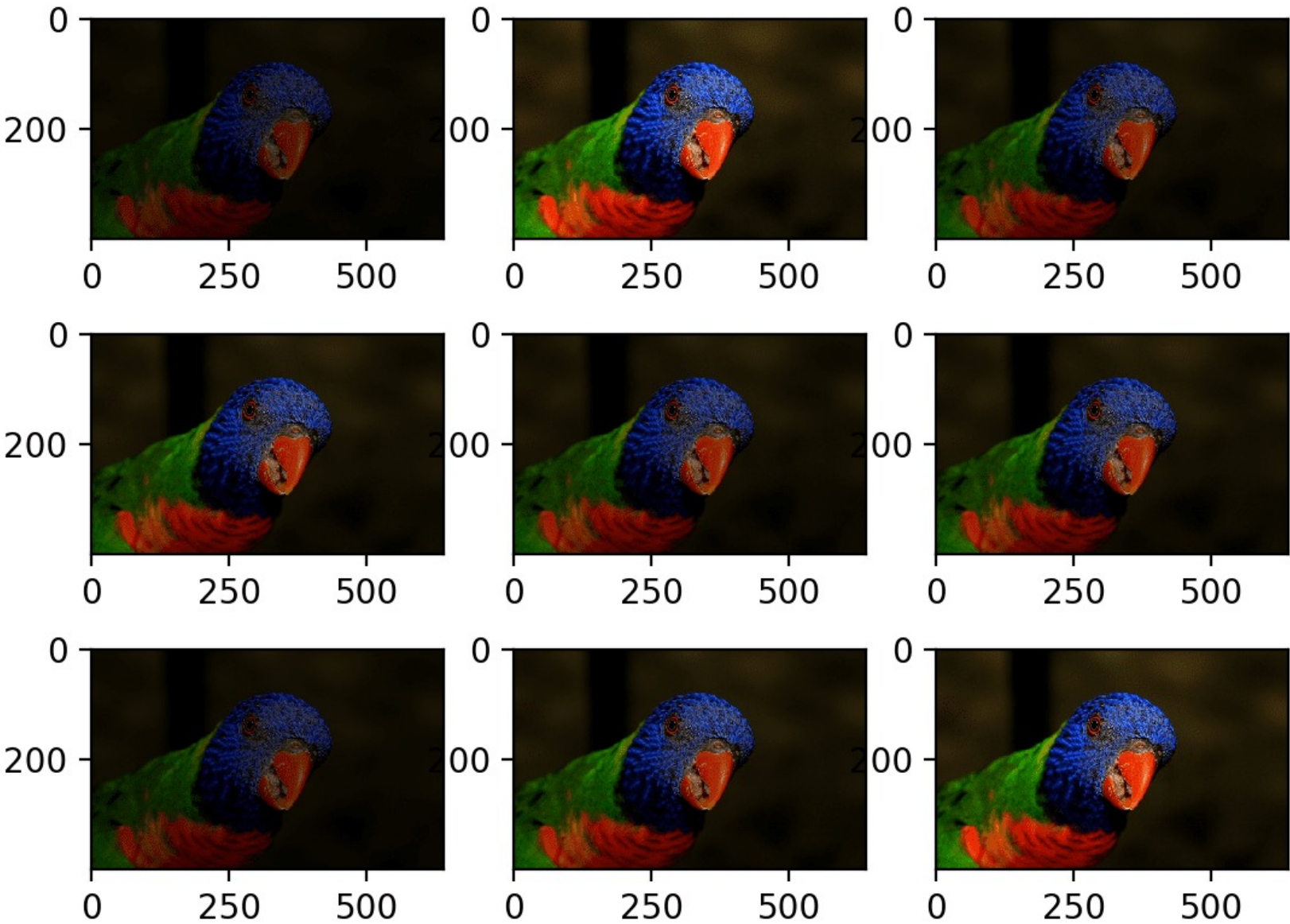
Data Augmentation



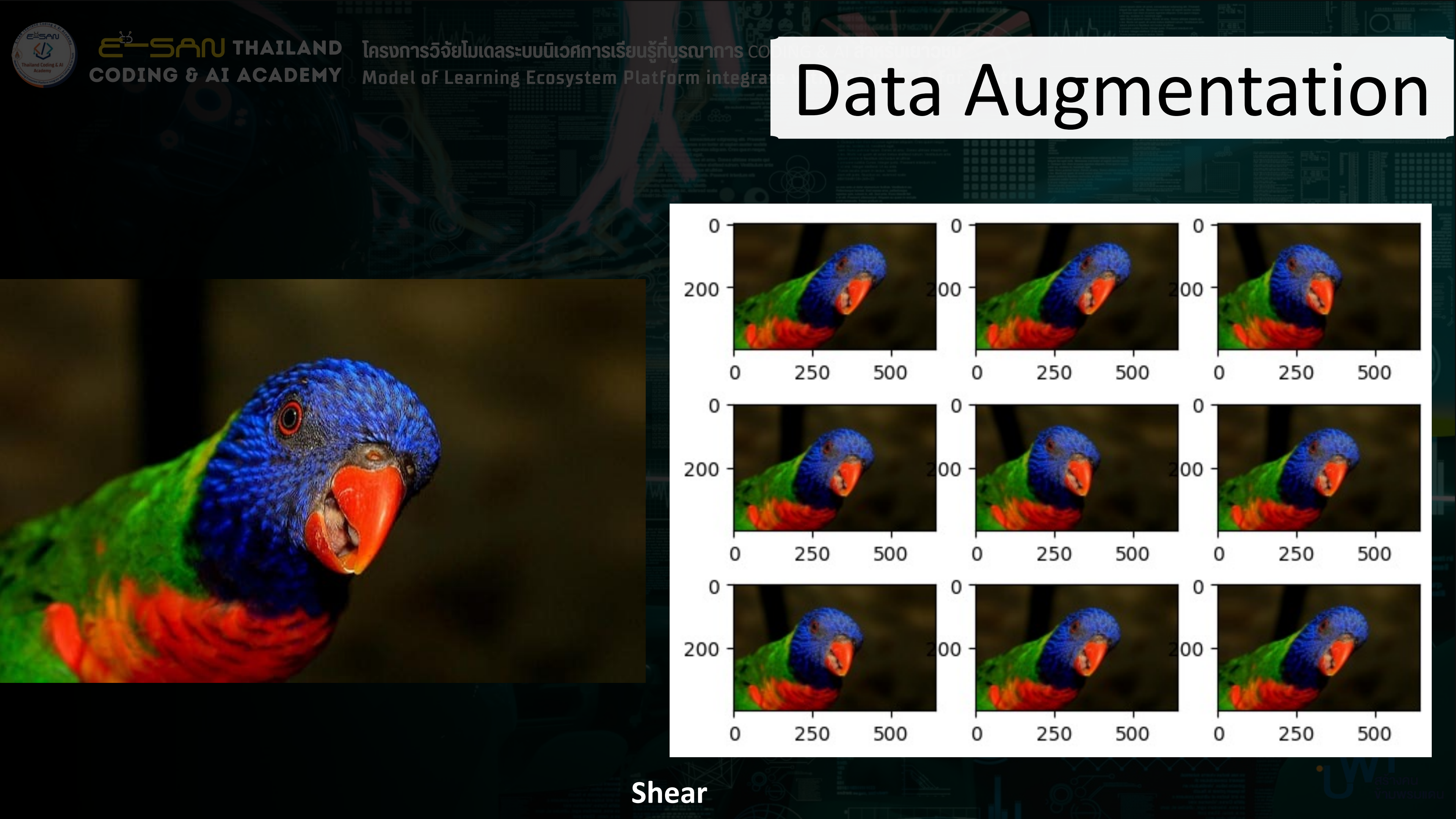
Rotation 30°



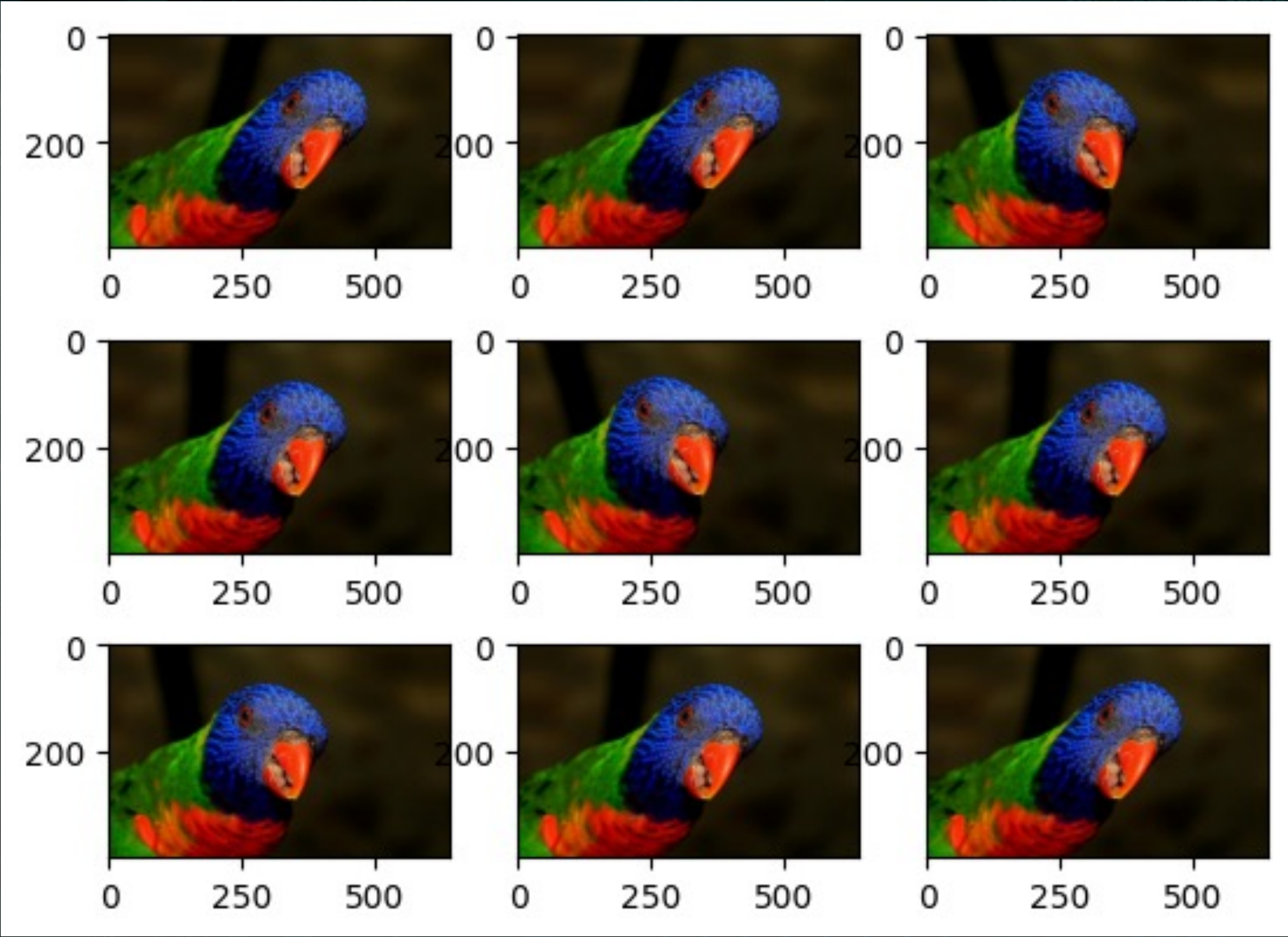
Data Augmentation



Bright



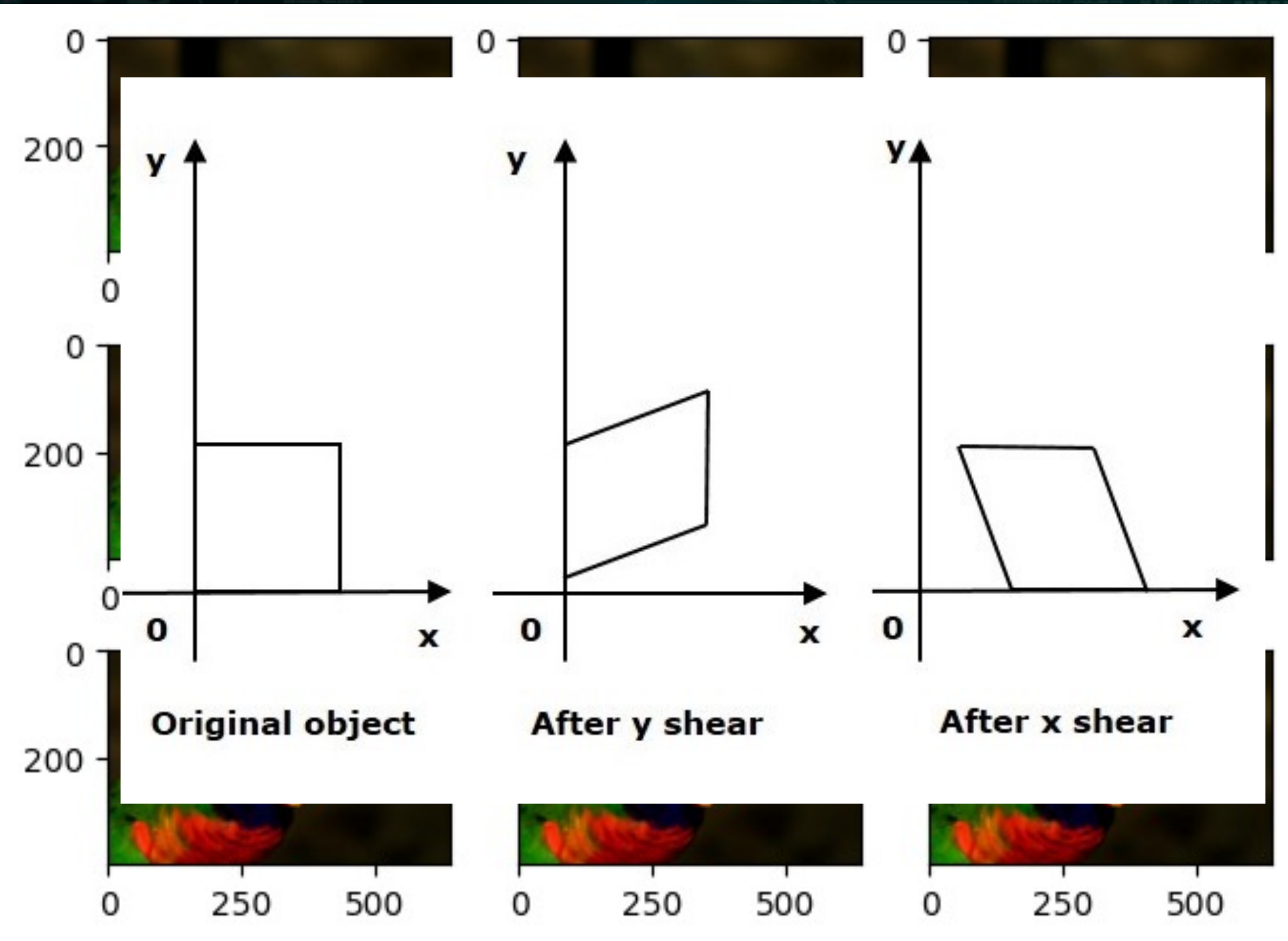
Data Augmentation



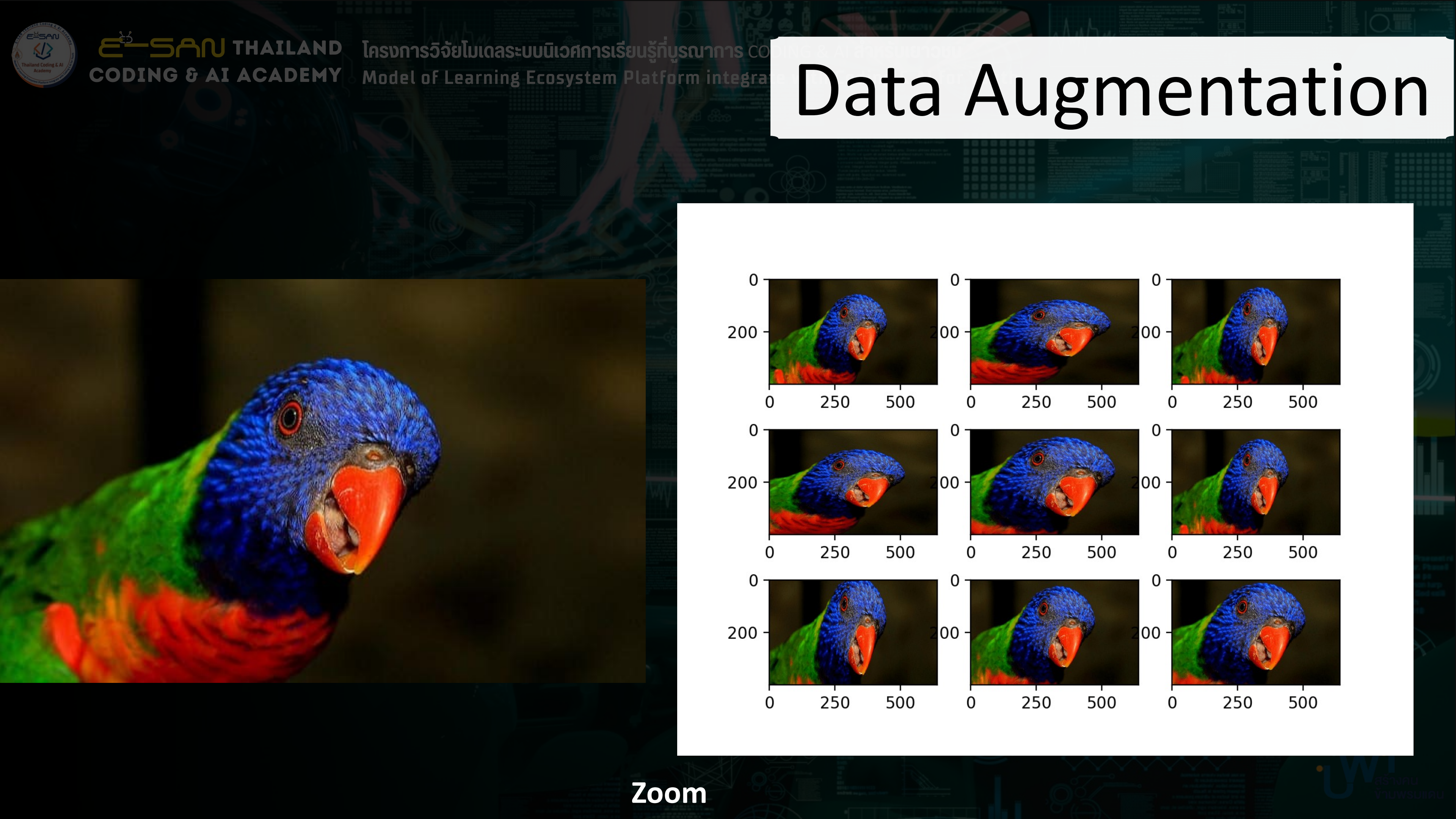
Shear



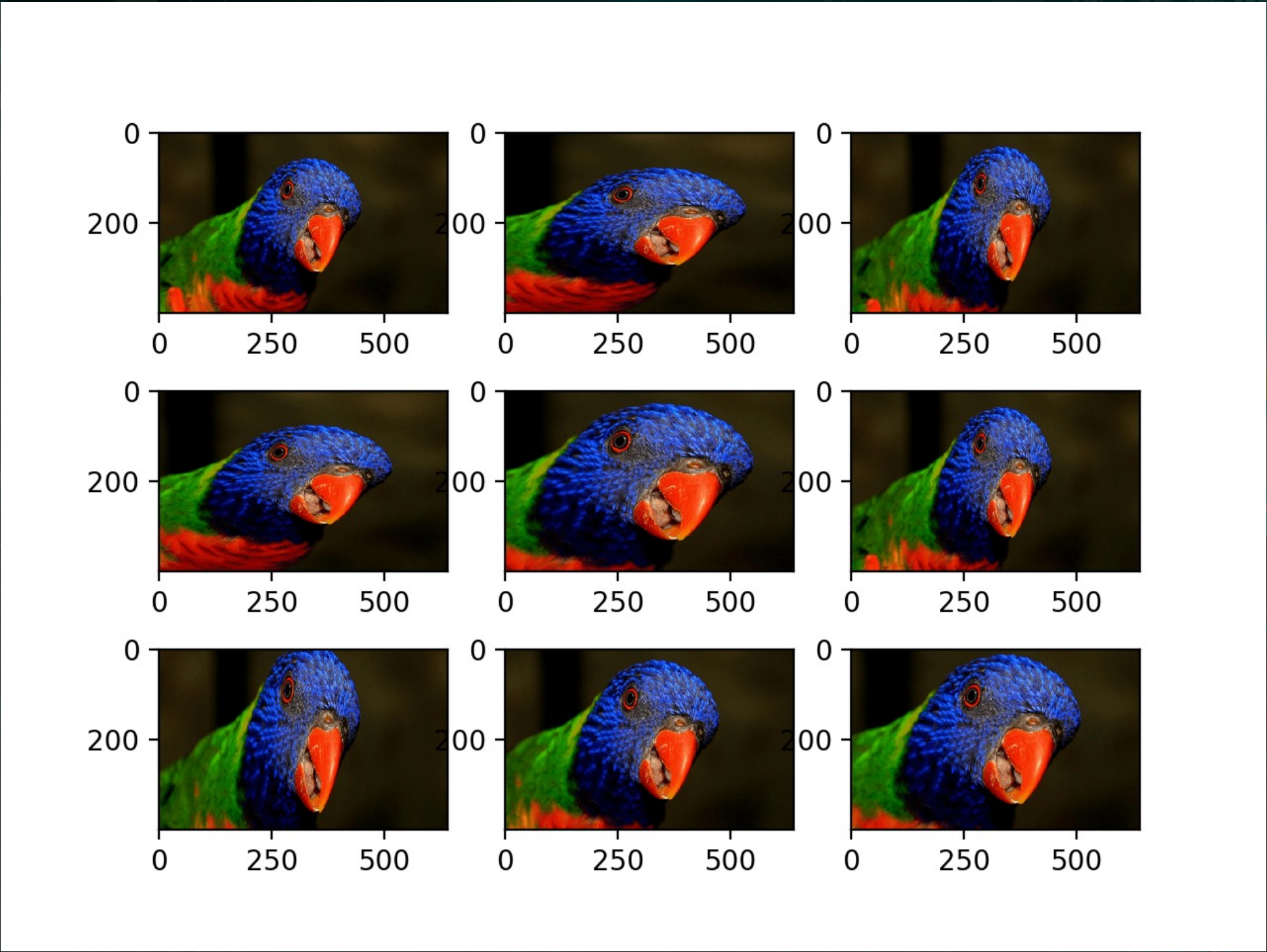
Data Augmentation



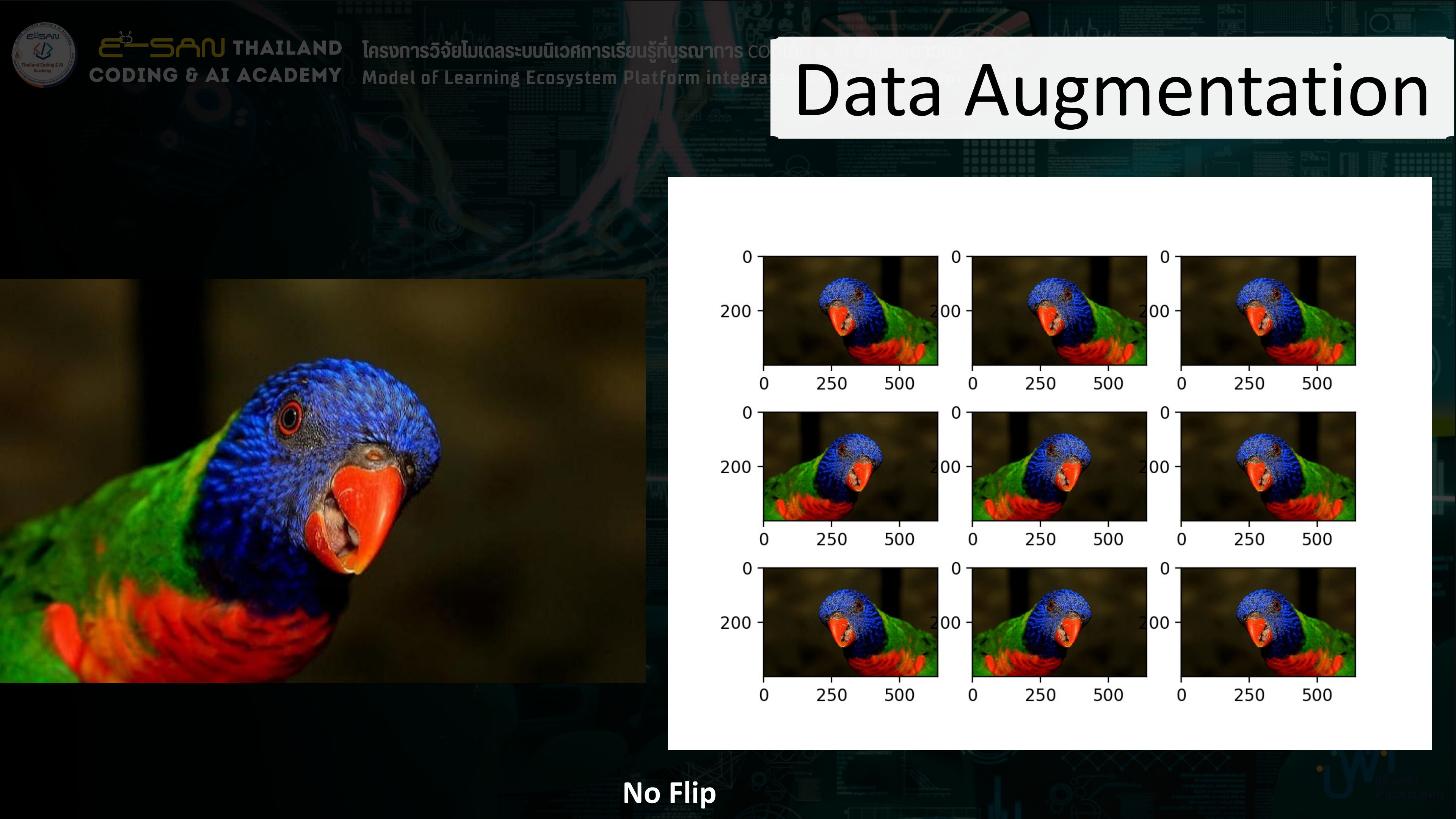
Shear



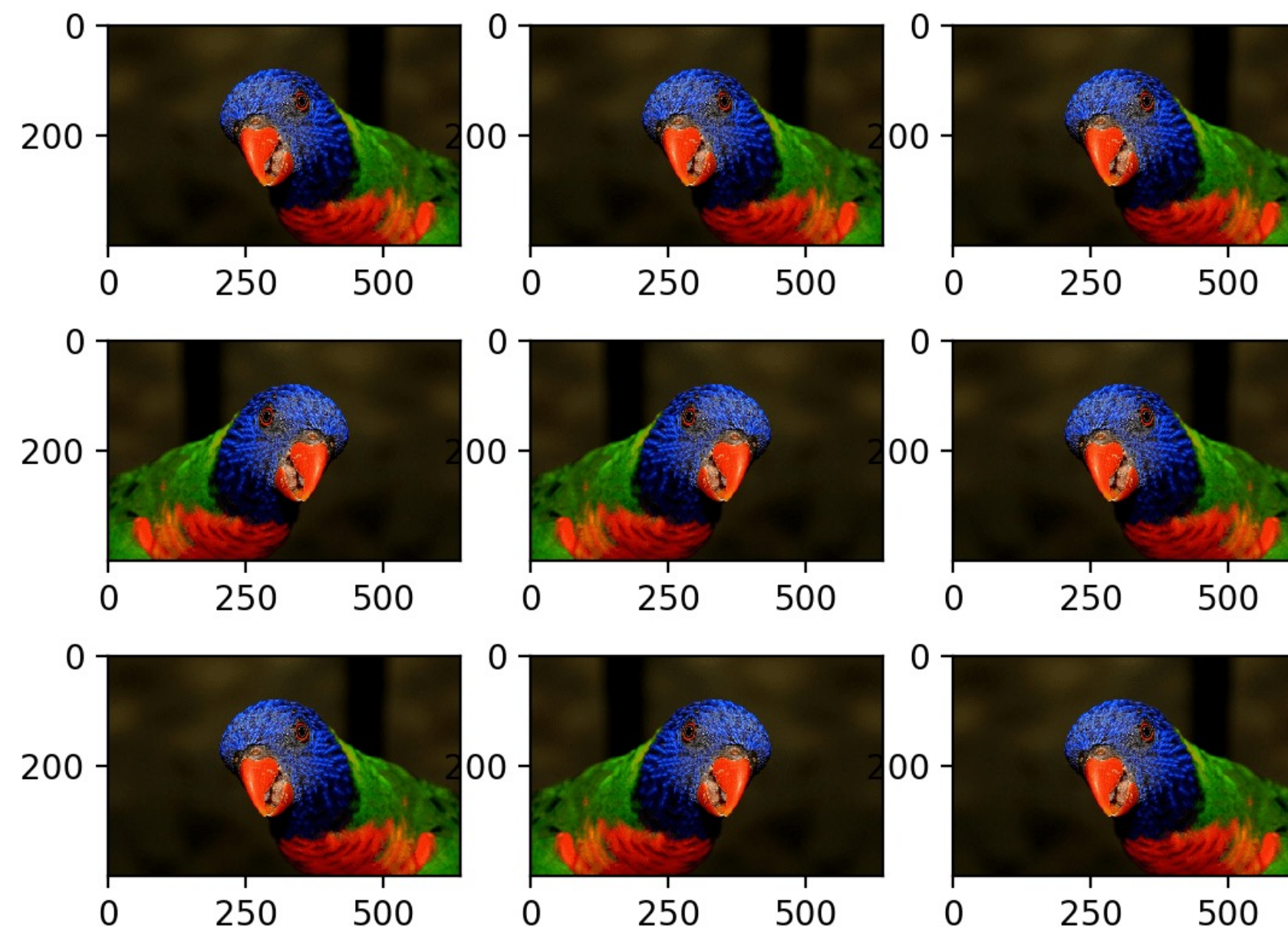
Data Augmentation



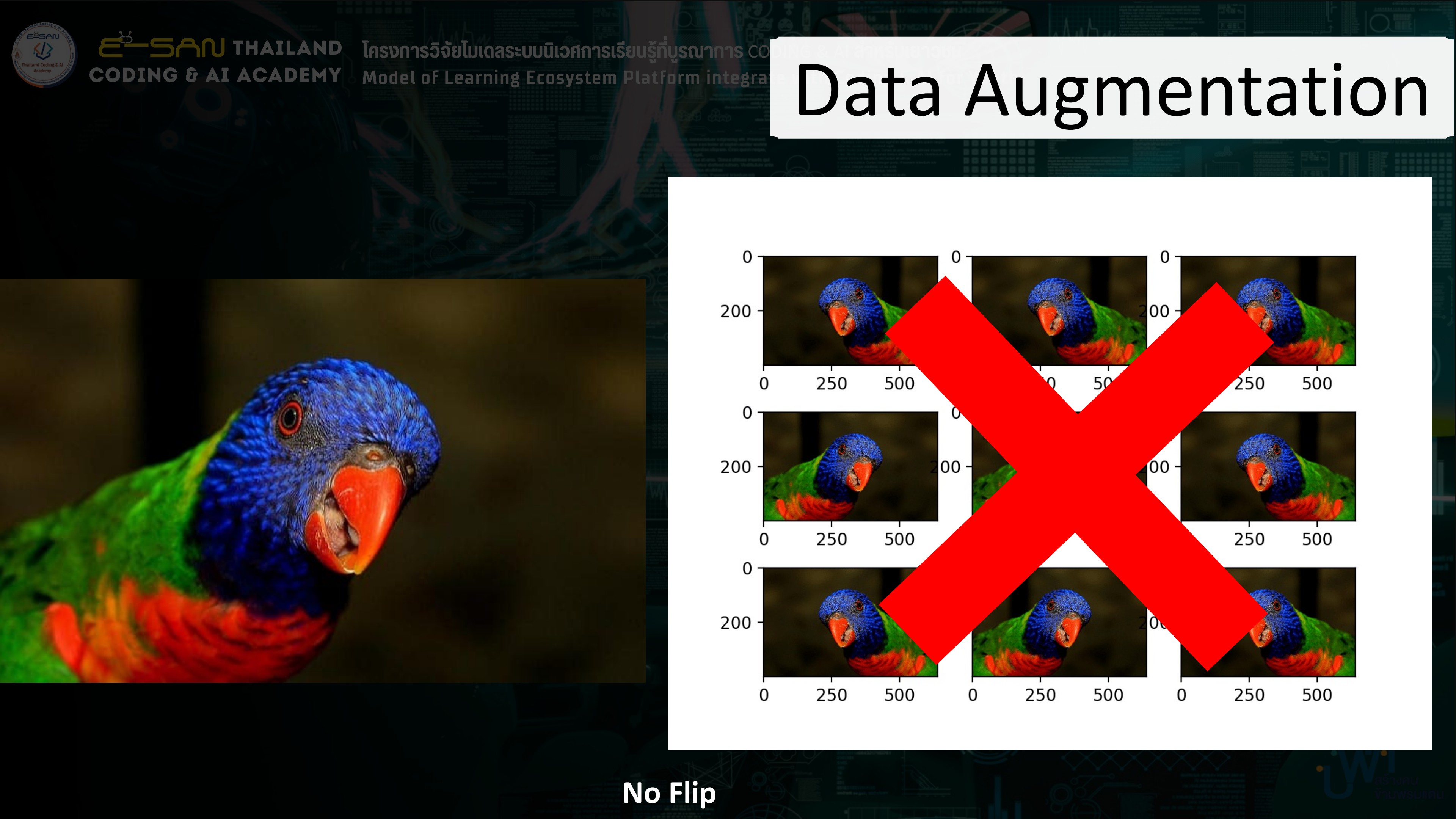
Zoom



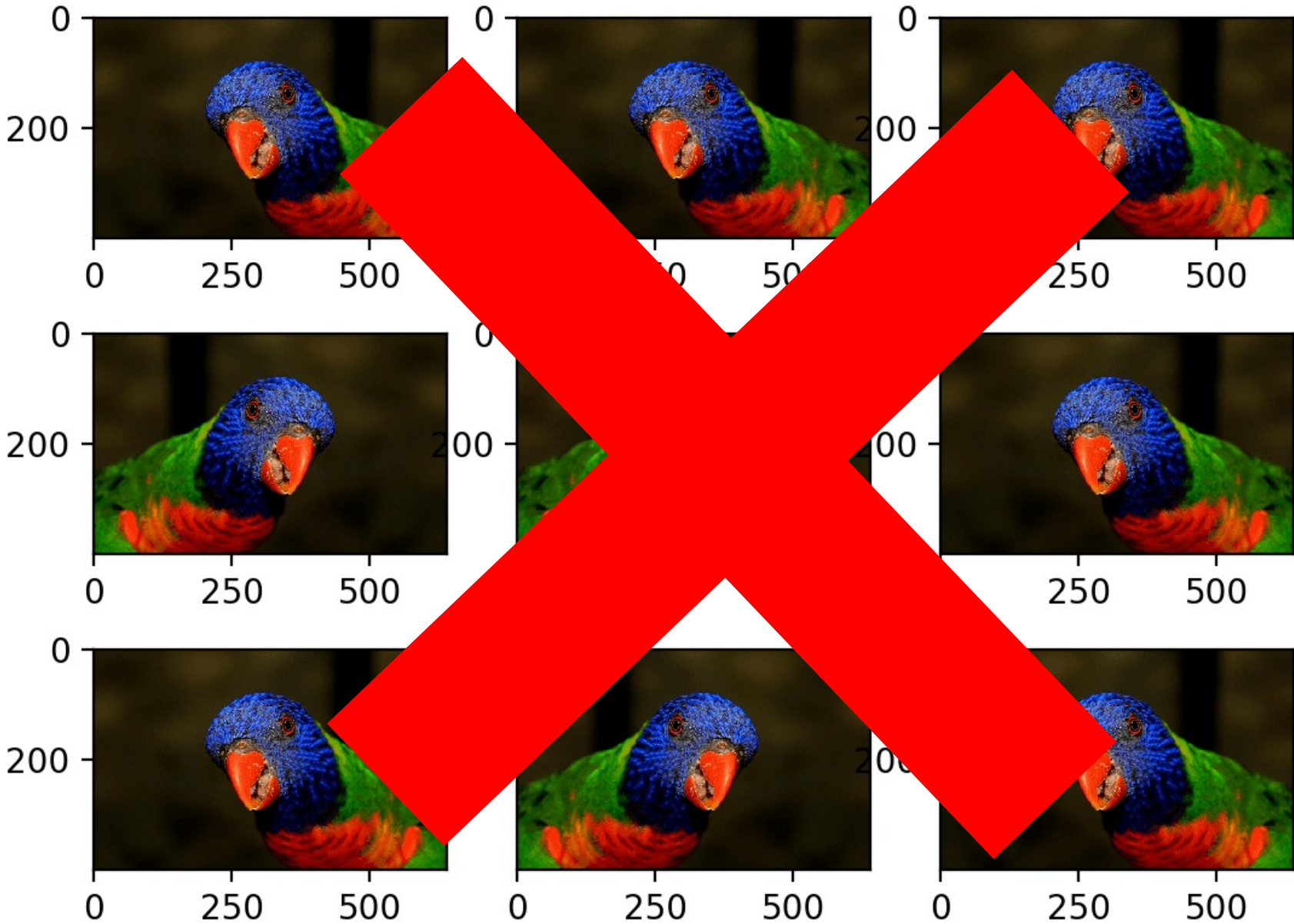
Data Augmentation



No Flip




Data Augmentation



No Flip

Data Augmentation

<https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/>

 **Machine Learning Mastery**
Making Developers Awesome at Machine Learning

Click to Take the FREE Computer Vision Crash-Course

Search...

Get Started Blog Topics ▾ EBooks FAQ About Contact

How to Configure Image Data Augmentation in Keras

by Jason Brownlee on July 5, 2019 in Deep Learning for Computer Vision 237






 Tweet  Tweet  Share  Share

Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset.






Training deep learning neural network models on more data can result in more skillful models, and the augmentation techniques can create variations of the images that can improve the ability of the fit models to generalize what they have learned to new images.

The Keras deep learning neural network library provides the capability to fit models using image data augmentation via the *ImageDataGenerator* class.

In this tutorial, you will discover how to use image data augmentation when training deep learning neural networks.

 **Welcome!**
I'm Jason Brownlee PhD
and I help developers get
results with machine learning.
[Read more](#)

Never miss a tutorial:

```
1 # example of horizontal shift image augmentation
2 from numpy import expand_dims
3 from keras.preprocessing.image import load_img
4 from keras.preprocessing.image import img_to_array
5 from keras.preprocessing.image import ImageDataGenerator
6 from matplotlib import pyplot
7 # load the image
8 img = load_img('bird.jpg')
9 # convert to numpy array
10 data = img_to_array(img)
11 # expand dimension to one sample
12 samples = expand_dims(data, 0)
13 # create image data augmentation generator
14 datagen = ImageDataGenerator(width_shift_range=[-200,200])
15 # prepare iterator
16 it = datagen.flow(samples, batch_size=1)
17 # generate samples and plot
18 for i in range(9):
19     # define subplot
20     pyplot.subplot(330 + 1 + i)
21     # generate batch of images
22     batch = it.next()
23     # convert to unsigned integers for viewing
24     image = batch[0].astype('uint8')
25     # plot raw pixel data
26     pyplot.imshow(image)
27     # show the figure
28     pyplot.show()
```

Coding Reference

