

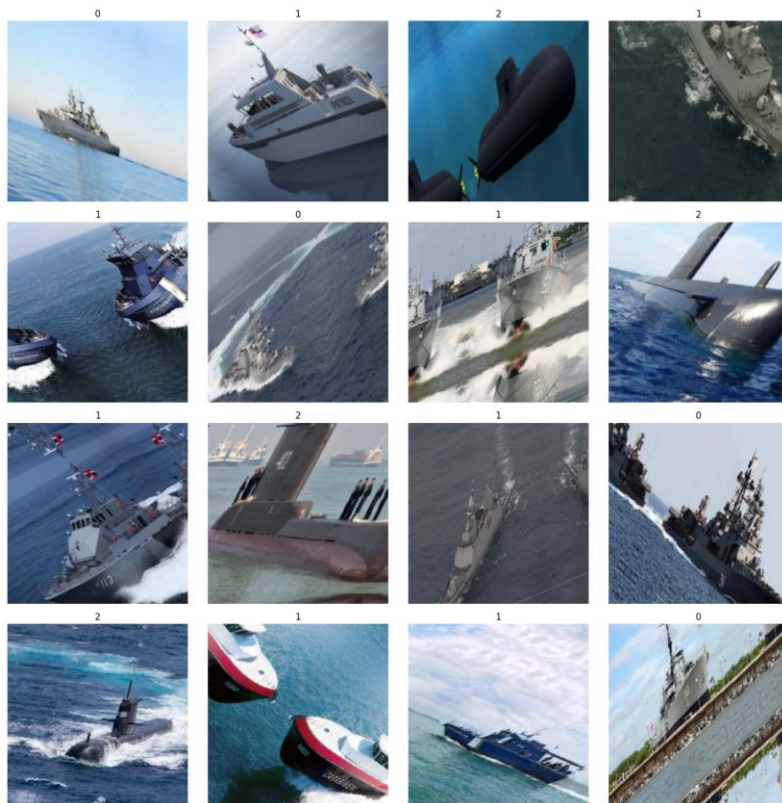
Lab7_1

```
def load_data(path):  
    ### START CODE HERE ###  
    transform = transforms.Compose([  
        transforms.Resize((224, 224)),  
        transforms.Pad(223, padding_mode='reflect'),  
        transforms.RandomAffine(degrees=45, translate=(0.1, 0.1), scale=(0.8, 1.2), shear=45),  
        transforms.CenterCrop(224),  
        transforms.ToTensor(),  
    ])  
  
    dataset = ImageFolder(path, transform=transform)  
  
    data_loader = DataLoader(dataset, batch_size=16, shuffle=True, num_workers=16, pin_memory=True)  
  
    return data_loader  
  
    ### END CODE HERE ###
```

ทำการ Image Transformation โดยจะทำการปรับภาพขนาดภาพเป็น (224,224), เพิ่มขนาดpadding 223 โดยเติมขอบด้วยmode reflect, สุ่มหมุนภาพ+ไม่เกิน45 องศา translationแนวนอน และ แนวตั้งไม่เกิน10%, crop ตรงกลางภาพให้ได้ขนาด(224, 224) และปรับภาพเป็น tensor ให้ค่าของแต่ละpixel อยู่ระหว่าง0 ถึง 1

โดยจะโหลดdataset จะใช้ImageFolder หลังจากนั้นจะนำdataset นั้นไปใช้Dataloader

```
def imshow_grid(images, labels):  
    fig, axes = plt.subplots(4, 4, figsize=(15, 15))  
    axes = axes.flatten()  
  
    for img, label, ax in zip(images, labels, axes):  
        img_np = img.permute(1, 2, 0).numpy()  
        ax.imshow(np.clip(img_np, 0, 1))  
        ax.set_title(f'{label.item()}')  
        ax.axis('off')  
  
    plt.tight_layout()  
    plt.show()
```



ฟังก์ชันนี้เป็นการสร้างตารางขนาด 4*4 มาเก็บภาพที่ผ่านการtransform มาแล้วโดยจะแสดงเลขlabel ของชนิดเรือตามตำแหน่งของfolder

```

class CustomImageDataset(Dataset):
    def __init__(self, data_dir, pad=None, padding_mode=None, gauss_noise=False, gauss_blur_ks=None, rotate=None,
                 self.p = p
                 self.resize = resize
                 self.pad = pad
                 self.padding_mode = padding_mode
                 self.gauss_noise = gauss_noise
                 self.gauss_blur_ks = gauss_blur_ks
                 self.rotate = rotate
                 self.shear = shear
                 self.center_crop = center_crop
                 self.data_dir = data_dir
                 self.dataset = ImageFolder(self.data_dir)
                 self.image_files = self.dataset.imgs

    def add_gaussian_blur(self, image):
        kernel = random.choice(self.gauss_blur_ks) if self.gauss_blur_ks else 5
        blur_image = cv2.GaussianBlur(image, (kernel, kernel), 0)
        return blur_image

    def add_gaussian_noise(self, image):
        noise_image = random_noise(image, var=0.02, mean=0, mode='gaussian')
        return noise_image

    def normalize(self, image):
        return (image - image.min()) / (image.max() - image.min())

```

```

    def center_crop_image(self, image, size):
        h, w = image.shape[:2]
        startx = w // 2 - (size[1] // 2)
        starty = h // 2 - (size[0] // 2)
        return image[starty:starty + size[0], startx:startx + size[1]]

    def magic_number(self):
        return random.random() < self.p

    def __getitem__(self, idx):
        image_path, label = self.image_files[idx]

        image = cv2.cvtColor(cv2.imread(image_path), cv2.COLOR_BGR2RGB)

        if self.gauss_noise:
            image = self.add_gaussian_noise(image)

        if self.gauss_blur_ks:
            image = self.add_gaussian_blur(image)

        if self.resize:
            image = cv2.resize(image, (self.resize, self.resize))

        if self.pad:
            image = cv2.copyMakeBorder(image, self.pad, self.pad, self.pad, self.pad, cv2.BORDER_CONSTANT if self.

        if self.rotate and self.magic_number():
            angle = random.randint(self.rotate[0], self.rotate[1])
            rows, cols, _ = image.shape
            M = cv2.getRotationMatrix2D((cols / 2, rows / 2), angle, 1)
            image = cv2.warpAffine(image, M, (cols, rows))

```

Class CustomImageDataset เอาไว้สำหรับการcustom image โดยจะมีฟังก์ชัน `__getitem__` เป็นฟังก์ชันที่เอาไว้เรียกการฟังก์ชันในการ custom image ตามในแบบที่เราต้องการ โดยจะมีการcustom image แบบ การเบลอด้วย Gaussian, การเพิ่มสัญญาณรบกวน (Gaussian noise), การปรับขนาด (resize), การเติมขอบ (padding), การหมุนภาพ (rotation), และการตัดส่วนกลาง (center cropping)