Tales Araujo Leonidas
Professor Patricia McManus
ITAI 2376: Deep Learning
April 21, 2024

Finding Descriptors and Creating Plots on Image Data: Reflective Journal

Introduction

This journal describes my experience performing Lab 1 of the AWS Academy course "*Application of Deep Learning to Text and Image Data*", Module 3. The Lab introduces the basics of Computer Vision, the technology that empowers computers to interpret visual information. There, I learned how to open, inspect, and extract features from image data using Python. The goal is to get familiarized with working with images while enhancing skills in handling and understanding them.

Experience Description

As usual, the Lab started installing the required frameworks and importing the necessary libraries, which mainly relied on *Pytorch*. Following, I loaded the *CIFAR10* train and test datasets and transformed them into tensors using *torchvision,* a library that provides tools and datasets for working with image data. After inspecting a random image and printing its shape, I had some key information about the *CIFAR10* dataset, as depicted and explained on the image to the side.

```
torch.Size([3, 32, 32]) Label:  2
```

The output above tells you 4 things:

1. This is a color image with **3** channels
2. The height of the image is **32** px.
3. The width of the image is **32** px.
4. Time image lable is **2**

To access individual images or batches of images and their labels easily, a *TensorDataset* is used. "To create a TensorDataset you need to pass the images (data_tensor) and labels (target_tensor) into Dataloader." The code snippet below exemplifies the process of creating a *TensorDataset* using the 50 first images of each split and then passing it into batches to the *dataloader*.

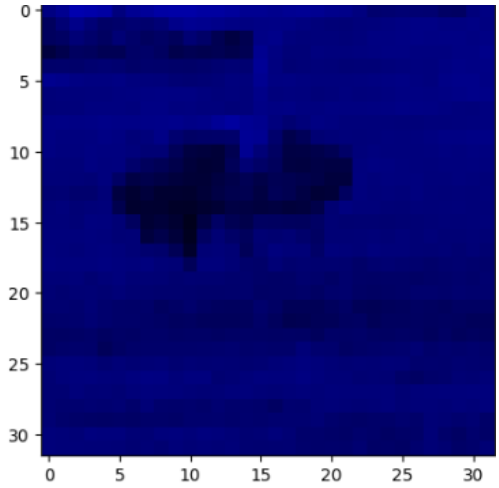
```
data_tensor = torch.Tensor(img_train.data[:50])
target_tensor = torch.Tensor(img_train.targets[:50])

tensor_dataset = TensorDataset(data_tensor, target_tensor)
```

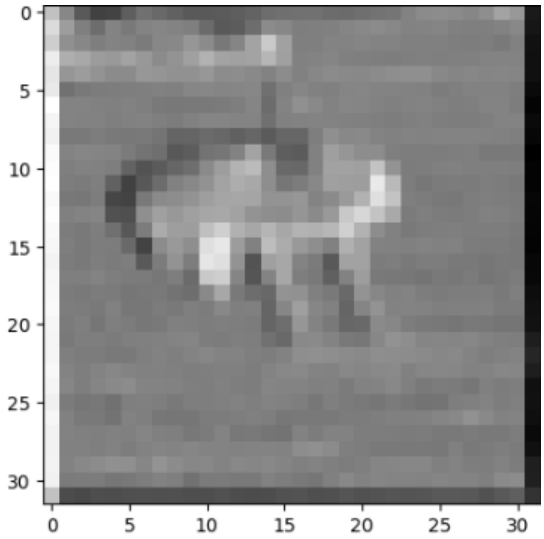
```
dataloader = torch.utils.data.DataLoader(
    tensor_dataset, batch_size=32, shuffle=True, drop_last=False
```

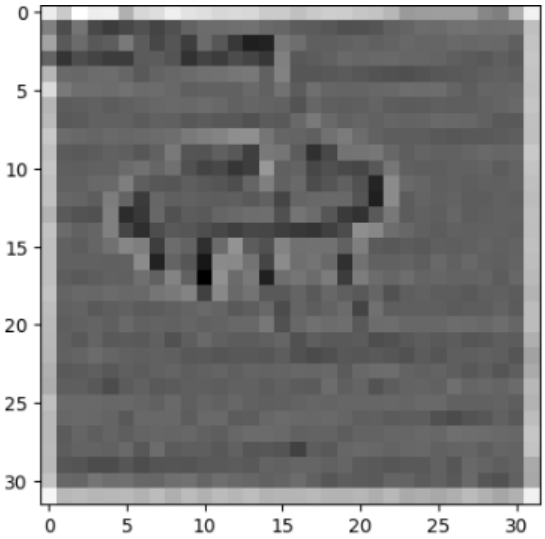
After converting the image object to an array to look and

manipulate the pixel values, the tensor was rearranged to the *matplotlib* expected format for displaying images (height, width, channels), and channels 1 and 2 (green and blue) were zeroed out to show the red color channel. The equivalent was applied in the sequence to display the green and blue color channels.



To extract features from the image, a convolutional layer was initialized, and a Laplace filter was used. “The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection” (AWS Academy). The images below depict the difference between the initial transformation of the image to grayscale and the subsequent feature map resulting from the edge detection using the Laplace filter.





Personal Reflection

I found the exercise of opening and analyzing image data enlightening since it deepened my grasp of Computer Vision concepts and increased my confidence in handling and interpreting image data for practical applications. Utilizing *Pytorch* and *torchvision* to manipulate image tensors and exploring feature extraction techniques like the Laplace filter was seamless and a great learning experience.

Improvements and Learning

In this Lab, I learned that it is possible to see the number of color channels in the image and its dimensions (height and width) by calling "*image.shape*".

```
# Print the image and label at the 42nd index
image, label = img_train[42]
print(image.shape, "Label: ",label)
```

```
torch.Size([3, 32, 32]) Label:  2
```

The Lab taught me the significance of tensor manipulation for image analysis, evident when rearranging image data to fit the expected format for visualization. Additionally, the exercise of zeroing out specific color channels provided a great understanding of how different layers of data contribute to the overall composition of an image. I also appreciated the power of convolutional layers in highlighting image details.

Conclusion

In conclusion, I have gained essential skills that will undoubtedly benefit my future projects through practical exercises involving the manipulation and feature extraction of image data. This lab has solidified my technical capabilities in using tools such as *Pytorch* and *torchvision*, and sharpened my analytical skills in interpreting image datasets. I feel confident that I am equipped with the foundational knowledge to tackle more advanced applications in computer vision.

Works Cited

AWS Academy. “Application of Deep Learning to Text and Image Data: Module 3.” AWS Academy, 2024, https://awsacademy.instructure.com/login/canvas