FCV ASSIGNMENT 01

SUBMITTED BY: AYESHA ZIA(20K-0414)

QUESTION 01/02/03

Ayesha Zia
20k-0414.

FCV Assignment-1 Date:

Q1)

1) (75, 67, 85)

Scaling

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 75 \\ 67 \\ 85 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 75 \\ 134 \\ 255 \\ 1 \end{bmatrix}$$

Rotation

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos 45 & -\sin 45 & 0 & 0 \\ \sin 45 & \cos 45 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix}$$

Translation

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 6 \\ 0 & 1 & 0 & 7 \\ 0 & 0 & 1 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 33.6 \\ 74 \\ 93 \\ 1 \end{bmatrix}$$

2) (100, 200, 500)

Scaling

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 100 \\ 200 \\ 500 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 400 \\ 1000 \\ 3000 \\ 1 \end{bmatrix}$$

Rotation

$$\begin{bmatrix} X_2 \\ Y_2 \end{bmatrix} = \begin{bmatrix} \cos 90 & -\sin 90 & 0 & 0 \\ \sin 90 & \cos 90 & 0 & 0 \end{bmatrix}$$

Date:

Translation

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 10 \\ 0 & 1 & 0 & 11 \\ 0 & 0 & 1 & 12 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 110 \\ 211 \\ 512 \\ 1 \end{bmatrix}$$

3) (345, 800, 560)

Scaling

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 11 & 0 & 0 \\ 0 & 0 & 12 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 345 \\ 800 \\ 560 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 3450 \\ 8800 \\ 6720 \\ 1 \end{bmatrix}$$

Rotation

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos 180 & -\sin 180 & 0 & 0 \\ \sin 180 & \cos 180 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 345 \\ 800 \\ 560 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} -345 \\ -800 \\ 0 \\ 1 \end{bmatrix}$$

Translation

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 345 \\ 800 \\ 560 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 349 \\ 805 \\ 566 \\ 1 \end{bmatrix}$$

4) (145, 1800, 360) Scaling

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 13 & 0 & 0 & 0 \\ 0 & 14 & 0 & 0 \\ 0 & 0 & 15 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 145 \\ 1800 \\ 360 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1885 \\ 25200 \\ 5400 \\ 1 \end{bmatrix}$$

Rotation

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos 360 & -\sin 360 & 0 & 0 \\ \sin 360 & \cos 360 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 145 \\ 1800 \\ 360 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 145 \\ 145 \\ 0 \\ 1 \end{bmatrix}$$

Date:

Translation

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 110 \\ 0 & 1 & 0 & 20 \\ 0 & 0 & 1 & 30 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 45 \\ 1800 \\ 360 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 155 \\ 1820 \\ 390 \\ 1 \end{bmatrix}$$

5) (45, 880, 460)

Scaling

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 15 & 0 & 0 & 0 \\ 0 & 16 & 0 & 0 \\ 0 & 0 & 17 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 45 \\ 880 \\ 460 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 675 \\ 14080 \\ 7820 \\ 1 \end{bmatrix}$$

Rotation

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos 75 & -\sin 75 & 0 & 0 \\ \sin 75 & \cos 75 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 45 \\ 880 \\ 460 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} -838.36 \\ 271.22 \\ 460 \\ 1 \end{bmatrix}$$

Translation

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 50 \\ 0 & 1 & 0 & 60 \\ 0 & 0 & 1 & 75 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 45 \\ 880 \\ 460 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 95 \\ 940 \\ 535 \\ 1 \end{bmatrix}$$

Q2) Rectangular → Spherical

1) (4, 5, 7)

$r = \sqrt{x^2+y^2+z^2}$

$= \sqrt{4^2+5^2+7^2} = \sqrt{16+25+49}$

$r = \sqrt{90} = 9.48683$

$\tan\theta = \frac{y}{x}$ $\quad \theta = \tan^{-1}\left(\frac{5}{4}\right)$

$\theta = 0.896$

$\Theta = \cos^{-1}\left(\frac{z}{\sqrt{x^2+y^2+z^2}}\right)$

$= \left(\frac{7}{\sqrt{90}}\right) = 1.028$

(9.4868, 0.896, 1.028)

Date: _____

2) (20, 40, 70)

$r = \sqrt{20^2 + 40^2 + 70^2}$

$= 83.06$

$\theta = \tan^{-1}\left(\frac{40}{20}\right)$

$= 63.43$

$\theta = \cos^{-1}\left(\frac{70}{83.06}\right)$

$= 1.076$

(83.06, 63.43, 1.076)

3) (30, 15, 65)

$r = \sqrt{30^2 + 15^2 + 65^2}$

$= 72.83$

$\theta = \tan^{-1}\left(\frac{15}{30}\right)$

$= 26.56$

$\theta = \cos^{-1}\left(\frac{65}{72.83}\right)$

$= 1.418$

(72.83, 26.56, 1.42)

4) (10, 115, 165)

$r = \sqrt{10^2 + 115^2 + 165^2}$

$= 193.04$

$\theta = \tan^{-1}(115/10)$

$= 1.489$

$\theta = \cos^{-1}\left(\frac{165}{193.04}\right) = 1.229.$

(193.04, 1.489, 1.229)

Spherical → Rectangular

1) (6, 45, 68)

$x = r\sin\theta\cos\theta$

$= 6\sin 45 \cos 68 = 2.569$

$y = r\sin\theta\sin\theta = 6\sin 45 \sin 68$

$= 4.005$

$z = r\cos\theta = 6\cos 45 = 4.337$

(2.569, 4.005, 4.337)

2) (20, 25, 58)

$x = 20\sin 25\cos 58 = 4.479$ ~~12.304~~

$y = 20\sin 25 \sin 58 = 7.168$

$z = 20\cos 25 = 18.126$

(4.479, 7.168, 18.126)

3) (10, 35, 78)

$x = 10\sin 35 \cos 78 = 1.192$

$y = 10\sin 35 \sin 78 = 5.61042$

$z = 10\cos 35 = 8.19152$

(1.192, 5.61, 8.191)

4) (110, 15, 18)

$x = 110\sin 15 \cos 18 = 27.08$

$y = 110\sin 15 \sin 18 = 8.79$

$z = 110\cos 15 = 106.25$

(27.08, 8.79, 106.25)

Date:

Q3)

A)

using fourier transform,

$F(u) = \int f(x) e^{-i 2\pi} ux \, dx$.

u = spatial frequency

f(u) = frequency domain.

rectangular pulse sos-

$f(x) = r(x/2)$

$r(x) = 1$ for $|x| < \frac{1}{2}$ and 0 otherwise.

After substitution,

$F(u) = \int r\left(\frac{x}{2}\right) e^{-i 2\pi} ux \, dx$

$= 2\int r\left(\frac{x}{2}\right) \cos 2\pi ux \, dx$.

$F(u) = 2\,\text{sinc}(u)$ where

$\text{sinc}(u) = \sin(\pi u)/\pi u$.

B)

$F(u,v) = \iint f(x,y)\, e^{-i 2\pi(ux+vy)} dx\,dy$

$f(x,y) = r\left(\frac{x}{2}\right) * \left(\frac{y}{1}\right)$

$r(x) = 1$ for $|x| < 1/2$ and otherwise.

After substituting:-

$F(u,v) = \iint x\left(\frac{x}{2}\right) r\left(\frac{y}{1}\right) e^{-i 2\pi(ux+vy)} dx\,dy$.

$F(u,v) = 2\,\text{sinc}(\pi u)\,\text{sinc}(\pi v)$ where $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$

QUESTION 04

A)

The MNIST contains a collection of 70,000 grayscale images of handwritten digits from 0 to 9, each of size 28x28 pixels. The dataset is divided into 60,000 training images and 10,000 test images. The MNIST dataset is often used as a benchmark for evaluating machine learning algorithms, particularly in the field of deep learning.

Many popular deep learning frameworks, such as TensorFlow and Keras, provide built-in tools for loading and working with the MNIST dataset. Keras, a popular deep learning framework, provides a simple interface for loading and working with the MNIST dataset. The dataset is included as part of the Keras library.

The train_images and train_labels variables contain the training data, while test_images and test_labels contain the test data. The images are represented as numpy arrays, where each pixel is an integer between 0 and 255 representing the grayscale intensity. The labels are represented as integers between 0 and 9, corresponding to the digit that each image represents.

The MNIST dataset has been used extensively in research on deep learning and computer vision, and has been used as the basis for many state-of-the-art image classification models. Because the dataset is relatively small and well-structured, it is a good starting point for exploring and experimenting with deep learning techniques.

One of the most popular uses of the MNIST dataset is in training and evaluating convolutional neural networks (CNNs), which have shown impressive results on the task of digit recognition. CNNs are a type of deep neural network that is particularly well-suited to image classification tasks, as they can learn to identify complex features and patterns within images.

In addition to CNNs, many other types of deep learning models have been trained on the MNIST dataset, including recurrent neural networks (RNNs), autoencoders, and generative adversarial networks (GANs). Researchers have also used the dataset to explore topics such as transfer learning, data augmentation, and model compression.

Overall, the MNIST dataset is a valuable resource for researchers and practitioners in the field of machine learning and computer vision, and continues to be a popular benchmark for evaluating new algorithms and techniques. While the dataset itself is relatively simple, it has played a key role in advancing the field of deep learning and has helped to pave the way for many of the breakthroughs that we see today.

B)

```
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from keras.utils import to_categorical
```

```
import matplotlib.pyplot as plt

# Load the MNIST dataset and preprocess the data
(train_images, train_labels), (test_images, test_labels) = mnist.load_data
()
train_images = train_images.reshape((60000, 28, 28, 1)) / 255.0
test_images = test_images.reshape((10000, 28, 28, 1)) / 255.0
train_labels = to_categorical(train_labels)
test_labels = to_categorical(test_labels)

# Define the CNN architecture
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dense(10, activation='softmax'))

# Compile the model and train it on the MNIST dataset
model.compile(optimizer='rmsprop', loss='categorical_crossentropy', metric
s=['accuracy'])
history = model.fit(train_images, train_labels, epochs=5, batch_size=64, v
alidation_data=(test_images, test_labels))

# Plot the training and validation accuracy over time
plt.plot(history.history['accuracy'], label='Training accuracy')
plt.plot(history.history['val_accuracy'], label='Validation accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```

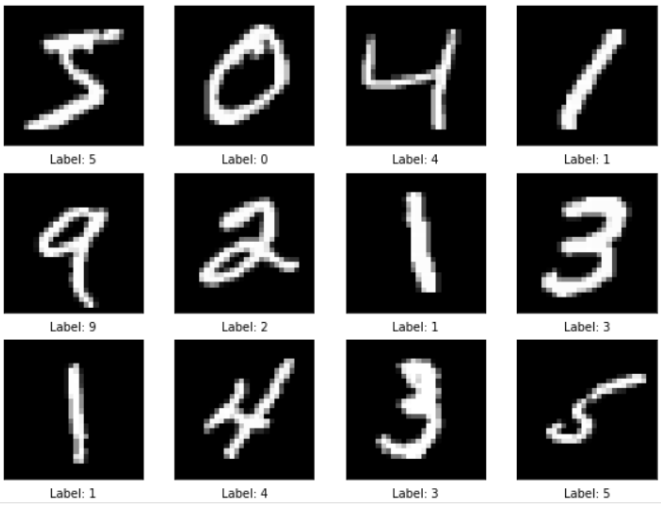
```
Epoch 1/5
938/938 [==============================] - 40s 41ms/step - loss: 0.1622 - accuracy: 0.9497 - val_loss: 0.0562 - val_accuracy: 0.9829
Epoch 2/5
938/938 [==============================] - 38s 41ms/step - loss: 0.0491 - accuracy: 0.9849 - val_loss: 0.0348 - val_accuracy: 0.9884
Epoch 3/5
```

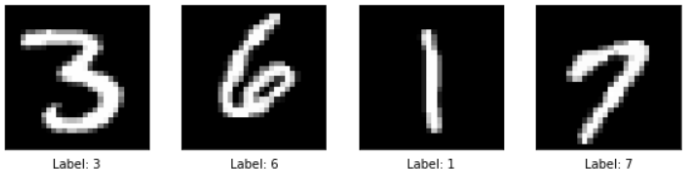
C)

```
from keras.datasets import mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()
print("Shape of x_train:", x_train.shape)
print("Shape of y_train:", y_train.shape)
```

```
print("Shape of x_test:", x_test.shape)
print("Shape of y_test:", y_test.shape)

import matplotlib.pyplot as plt

plt.figure(figsize=(10, 10))
for i in range(16):
    plt.subplot(4, 4, i+1)
    plt.imshow(x_train[i], cmap='gray')
    plt.xticks([])
    plt.yticks([])
    plt.xlabel("Label: " + str(y_train[i]))
plt.show()

import numpy as np

unique, counts = np.unique(y_train, return_counts=True)
print("Training Set Label Distribution:")
print(dict(zip(unique, counts)))

unique, counts = np.unique(y_test, return_counts=True)
print("Testing Set Label Distribution:")
print(dict(zip(unique, counts)))

#checking for any missing values in the dataset
print("Missing in x_train:", np.isnan(x_train).any())
print("Missing in y_train:", np.isnan(y_train).any())
print("Missing in x_test:", np.isnan(x_test).any())
print("Missing in y_test:", np.isnan(y_test).any())

#normalizing the dataset
x_train = x_train / 255.0
x_test = x_test / 255.0
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [==============================] - 0s 0us/step
Shape of x_train: (60000, 28, 28)
Shape of y_train: (60000,)
Shape of x_test: (10000, 28, 28)
Shape of y_test: (10000,)
```






```
Training Set Label Distribution:
{0: 5923, 1: 6742, 2: 5958, 3: 6131, 4: 5842, 5: 5421, 6: 5918, 7: 6265, 8: 5851, 9: 5949}
Testing Set Label Distribution:
{0: 980, 1: 1135, 2: 1032, 3: 1010, 4: 982, 5: 892, 6: 958, 7: 1028, 8: 974, 9: 1009}
Missing in x_train: False
Missing in y_train: False
Missing in x_test: False
Missing in y_test: False
```

QUESTION 05

Abstract

Our MNIST Digit Classification project aims to develop a neural network model capable of accurately identifying handwritten digits in the MNIST dataset. The dataset consists of 60,000 training images and 10,000 test images of handwritten digits from 0 to 9.

The project will follow a supervised learning approach, where the neural network will be trained using the labeled training data. The neural network architecture will consist of multiple layers, including an input layer, one or more hidden layers, and an output layer. The model will use backpropagation to optimize the weights and biases of the neurons in the network.

Various neural network architectures will be explored, including fully connected neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs), to determine the optimal architecture for the problem at hand. The model's performance will be evaluated using metrics such as accuracy, precision, recall, and F1 score.

The project will be implemented using Python and the Keras library. The Keras library provides a user-friendly interface to create and train neural networks efficiently. The project will also make use of visualization tools such as Matplotlib and TensorBoard to visualize the neural network's performance.

The successful completion of this project will result in a high-performing neural network model capable of accurately classifying handwritten digits. This model can be used in various applications, such as OCR systems and automated postal sorting.

The main objective of building this classifier is to train our neural network in such a way that it will be able to detect the input image. Our goal is to use these input images to develop AI-based approaches to predict and detect what particular digit is represented in the input image. Our input image will consist of digits from 0 to nine and our trained neural network will have to detect which digit is represented in the image. Our neural network is a single-digit detection neural network. If we are feeding a handwritten digit in this neural network, this trained neural network will successfully tell us what that digit is. Hence, once we train our neural network to detect the digit correctly, it is then tested with some test data. This whole process of training and testing will help us in building a single-digit detection classifier.

Features

Some dependencies needed are mentioned below

- The NumPy library will produce the array from the images
- Matplotlib, numpy arrays will be visualized using this library. Once we plot the array on matplotlib we can see which image we have.
- Seaborn is also used for visualization
- cv2 is an open computer vision library. One of the important libraries used for image recognition tasks.
- TensorFlow and Keras
- Randomseeds
- Python Imaging Library (PIL)
- Confusion matrix from TensorFlow.