14^{th} August 2023

Saakshi P- 1GA20CS120

20MAT55 (Linear Algebra for Machine Learning)
Assignment 2

1. Write a Python program to find the inverse of a matrix $A = \begin{bmatrix} 1 & -2 & 3 \\ 0 & 1 & -3 \\ 2 & -5 & 6 \end{bmatrix}$ using Gauss Jordan method.
2. Write a Python program for the following
   a. Convert two images in matrix form (say) A1 and A2
   b. Generate a new image with matrix operation A1+A2
   c. Generate a new image with matrix operation 0.5*A1+0.5*A2
   d. Display original and transformed images.
3. Write a Python program to generate Null Space and Column Space of the following matrix $A = \begin{bmatrix} -3 & 6 & -1 & 1 & -7 \\ 1 & -2 & 2 & 3 & -1 \\ 2 & -4 & 5 & 8 & -4 \end{bmatrix}$
4. Write a Python program to generate basis for Column Space of the following matrix $A = \begin{bmatrix} 1 & 2 & 3 & -4 & 8 \\ 1 & 2 & 0 & 2 & 8 \\ 2 & 4 & -3 & 10 & 9 \\ 3 & 6 & 0 & 6 & 9 \end{bmatrix}$

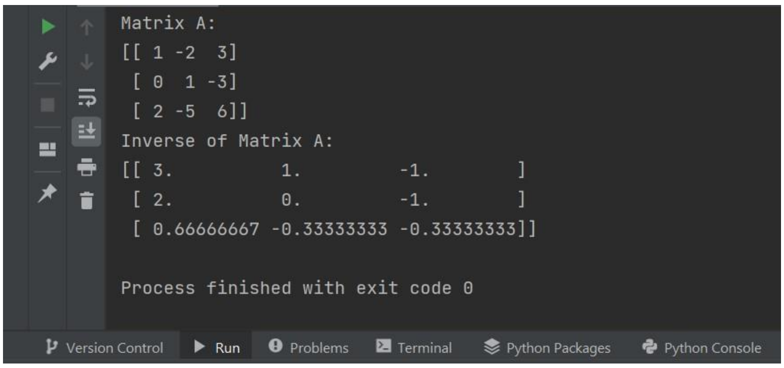
Q1. Write a Python program to find the inverse of a matrix A using Gauss Jordan method.

Ans.

```
import numpy as np

def gauss_jordan_inverse(matrix):
    n = len(matrix)
    augmented_matrix = np.hstack((matrix, np.identity(n)))

    for col in range(n):
        pivot = augmented_matrix[col, col]

        for row in range(col + 1, n):
```

```
            factor = augmented_matrix[row, col] / pivot
            augmented_matrix[row, :] -= factor * augmented_matrix[col, :]

    for col in range(n - 1, -1, -1):
        pivot        =        augmented_matrix[col,        col]
augmented_matrix[col, :] /= pivot

        for row in range(col - 1, -1, -1):
            factor = augmented_matrix[row, col]
            augmented_matrix[row, :] -= factor * augmented_matrix[col, :]

    inverse   =   augmented_matrix[:,   n:]
return inverse

# Define the matrix A matrix_A
= np.array([[1, -2, 3],
                [0, 1, -3],
                [2, -5, 6]])

# Calculate the inverse using Gauss-Jordan method inverse_A
= gauss_jordan_inverse(matrix_A)

print("Matrix A:") print(matrix_A)

print("Inverse of Matrix A:") print(inverse_A)
```

OUTPUT:



Q2. Write a Python program for the following

a. Convert two images in matrix form (say) A1 and A2

b. Generate a new image with matrix operation A1+A2

c. Generate a new image with matrix operation 0.5*A1+0.5*A2

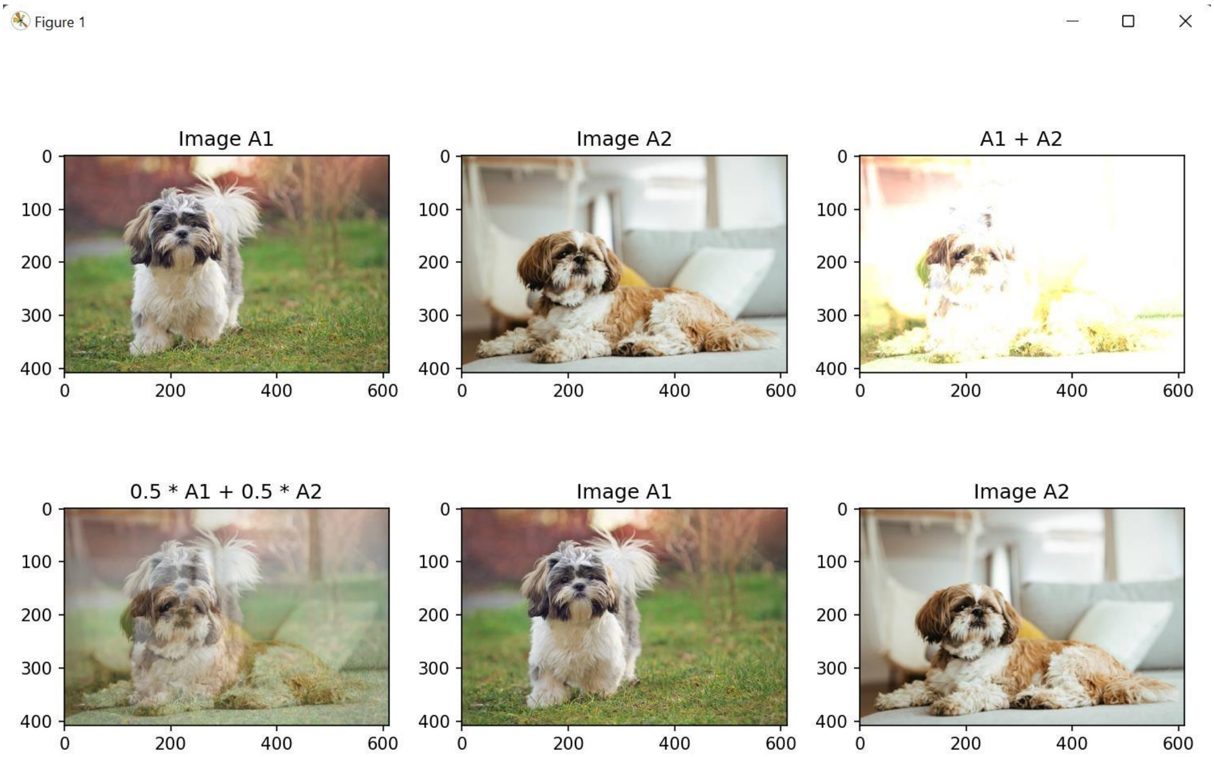
d. Display original and transformed images,

Ans.

```
import numpy as np import
matplotlib.pyplot as plt import
matplotlib.image as mpimg

# Load images A1 and A2 (replace these with your image paths)
A1 = mpimg.imread(r'C:\Users\Archana\Desktop\istockphoto-466754640-612x612.jpg')
A2 = mpimg.imread(r'C:\Users\Archana\Desktop\istockphoto-1218150093-612x612.jpg')
# Normalize pixel values to the range [0, 1]
A1 = A1 / 255.0
A2 = A2 / 255.0 #
Matrix operations
result_sum = A1 +
A2
result_weighted = 0.5 * A1 + 0.5 * A2

# Display original and transformed images plt.figure(figsize=(10,
8))

plt.subplot(2,  3,  1)
plt.imshow(A1)
plt.title('Image A1')

plt.subplot(2,  3,  2)
plt.imshow(A2)
plt.title('Image A2')

plt.subplot(2, 3, 3)
plt.imshow(result_sum)
plt.title('A1 + A2') plt.subplot(2, 3,
4) plt.imshow(result_weighted)
plt.title('0.5 * A1 + 0.5 * A2')
```

```
plt.subplot(2, 3, 5)
plt.imshow(A1)
plt.title('Image A1')

plt.subplot(2, 3, 6)
plt.imshow(A2)
plt.title('Image A2')
plt.tight_layout()
plt.show()
```

OUTPUT:



Q 3. Write a Python program to generate Null Space and Column Space of the matrix A

Ans.

```
import numpy as np

# Define the matrix A
```

```
matrix_A = np.array([[-3, 6, -1, 1, -7],
                     [1, -2, 2, 3, -1],
                     [2, -4, 5, 8, -4]])

# Calculate the null space using SVD (Singular Value Decomposition)
_, _, V = np.linalg.svd(matrix_A) null_space
= V[-1, :]

# Calculate the column space by taking a subset of the original matrix columns
column_space = matrix_A[:, :3]

print("Matrix A:") print(matrix_A)

print("Null Space of Matrix A:") print(null_space)

print("Column Space of Matrix A:") print(column_space)
```

OUTPUT:

```
Matrix A:
[[-3  6 -1  1 -7]
 [ 1 -2  2  3 -1]
 [ 2 -4  5  8 -4]]
Null Space of Matrix A:
[-0.43219587  0.43445438  0.55418604  0.23491278  0.5120058 ]
Column Space of Matrix A:
[[-3  6 -1]
 [ 1 -2  2]
 [ 2 -4  5]]

Process finished with exit code 0
```

Q 4. Write a Python program to generate basis for Column Space of the matrix Ans.

```
import numpy as np

# Define the matrix A
matrix_A = np.array([[1, 2, 3, -4, 8],
                     [1, 2, 0, 2, 8],
                     [2, 4, -3, 10, 9],
                     [3, 6, 0, 6, 9]])
```

# Perform Gaussian elimination to get the reduced row echelon form (rref) rref,
_ = np.linalg.qr(matrix_A)

# Extract the columns corresponding to the pivot columns to form a basis for column space
pivot_columns = np.where(rref[:, :-1].any(axis=0))[0] basis_column_space = matrix_A[:, pivot_columns]

print("Matrix A:") print(matrix_A)

print("Basis for Column Space of Matrix A:") print(basis_column_space)

OUTPUT: