**# Actual training loss data using Shanghai Tech Data Set**

import matplotlib.pyplot as plt

import numpy as np

# Placeholder data for demonstration

# Replace these with your actual training loss data

samples = np.linspace(0, 1000, 400)  # Simulated number of samples

loss1 = 20 / (samples + 1) + np.random.normal(0, 0.1, len(samples))  # Example for loss1

loss2 = 15 / (samples + 1) + np.random.normal(0, 0.1, len(samples))  # Example for loss2

loss = 10 / (samples + 1) + np.random.normal(0, 0.1, len(samples))   # Example for loss

# Plotting

plt.figure(figsize=(8, 6))

plt.plot(samples, loss1, color='green', label='loss1')

plt.plot(samples, loss2, color='purple', label='loss2')

plt.plot(samples, loss, color='maroon', label='loss')

# Labels and Title

plt.xlabel("No. of Samples into Training")

plt.ylabel("Training Loss")

plt.title("Training loss in 3 different phases")

plt.legend()

# Display the plot

plt.show()

**# Actual training loss data using UCSD Pedestrian Data Set**

import matplotlib.pyplot as plt

import numpy as np

# Placeholder data for demonstration

# Replace these with your actual training loss data

samples = np.linspace(0, 2500, 800)  # Simulated number of samples

loss1 = 20 / (samples + 1) + np.random.normal(0, 0.1, len(samples))  # Example for loss1

loss2 = 15 / (samples + 1) + np.random.normal(0, 0.1, len(samples))  # Example for loss2

loss = 10 / (samples + 1) + np.random.normal(0, 0.1, len(samples))   # Example for loss

# Plotting

plt.figure(figsize=(8, 6))

plt.plot(samples, loss1, color='green', label='loss1')

plt.plot(samples, loss2, color='purple', label='loss2')

plt.plot(samples, loss, color='maroon', label='loss')

# Labels and Title

plt.xlabel("No. of Samples into Training")

plt.ylabel("Training Loss")

plt.title("Training loss in 3 different phases")

plt.legend()

# Display the plot

plt.show()

**# Actual training loss data using User Collected Dataset**

import matplotlib.pyplot as plt

import numpy as np

# Placeholder data for demonstration

# Replace these with your actual training loss data

samples = np.linspace(0, 55000, 6000)  # Simulated number of samples

loss1 = 20 / (samples + 1) + np.random.normal(0, 0.1, len(samples))  # Example for loss1

loss2 = 15 / (samples + 1) + np.random.normal(0, 0.1, len(samples))  # Example for loss2

loss = 10 / (samples + 1) + np.random.normal(0, 0.1, len(samples))   # Example for loss

# Plotting

plt.figure(figsize=(8, 6))

plt.plot(samples, loss1, color='green', label='loss1')

plt.plot(samples, loss2, color='purple', label='loss2')

plt.plot(samples, loss, color='maroon', label='loss')

# Labels and Title

plt.xlabel("No. of Samples into Training")

plt.ylabel("Training Loss")

plt.title("Training loss in 3 different phases")

plt.legend()

# Display the plot

plt.show()