Image classification using PyTorch typically involves the following steps:

1. **Data Loading:** Load and preprocess the image dataset. PyTorch provides tools like **torchvision** for handling common image datasets.
2. **Model Definition:** Define a convolutional neural network (CNN) using PyTorch's **nn.Module** class. CNNs are well-suited for image classification tasks.
3. **Loss Function:** Choose a suitable loss function to measure the difference between predicted and actual labels. Common choices for classification tasks include CrossEntropyLoss.
4. **Optimizer:** Select an optimizer (e.g., SGD, Adam) to update the model parameters during training.
5. **Training Loop:** Iterate over the training dataset, forward pass the images through the network, calculate the loss, backpropagate the gradients, and update the model parameters.
6. **Validation/Test:** Evaluate the trained model on a separate validation or test dataset to assess its performance.