**Explanation**

We will use the Cats Faces Dataset which consists of more than 15,700 cats’ images.

Since generative modeling is an unsupervised learning method, hence there are no labels on the images.

The main difference between **nn.Conv2d** and **nn.ConvTranspose2d** in PyTorch lies in their operation and purpose within a neural network.

**nn.Conv2d:**

This is a convolutional layer commonly used for downsampling or extracting features from an input image. It performs a 2D convolution operation on the input tensor. It applies a set of learnable filters (kernels) to the input tensor and produces a feature map as output. The output tensor's spatial dimensions (width and height) are typically smaller than the input tensor, leading to downsampling or feature extraction.

**nn.ConvTranspose2d:**

This is a transposed convolutional layer or deconvolutional layer, often used for upsampling or generating new data from a learned representation. It performs an inverse operation of the convolution by transforming a lower-dimensional input into a higher-dimensional output. It applies a set of learnable filters (kernels) to the input tensor and produces an output tensor with larger spatial dimensions. This layer can be used in generative models like GANs to upsample feature maps and generate higher-resolution images.

In summary, **nn.Conv2d** is primarily used for downsampling or feature extraction, while **nn.ConvTranspose2d** is used for upsampling or generating new data. They serve different purposes in the network architecture, with **nn.ConvTranspose2d** typically employed in tasks such as image generation or semantic segmentation, where increasing the spatial dimensions is required.