In [ ]:

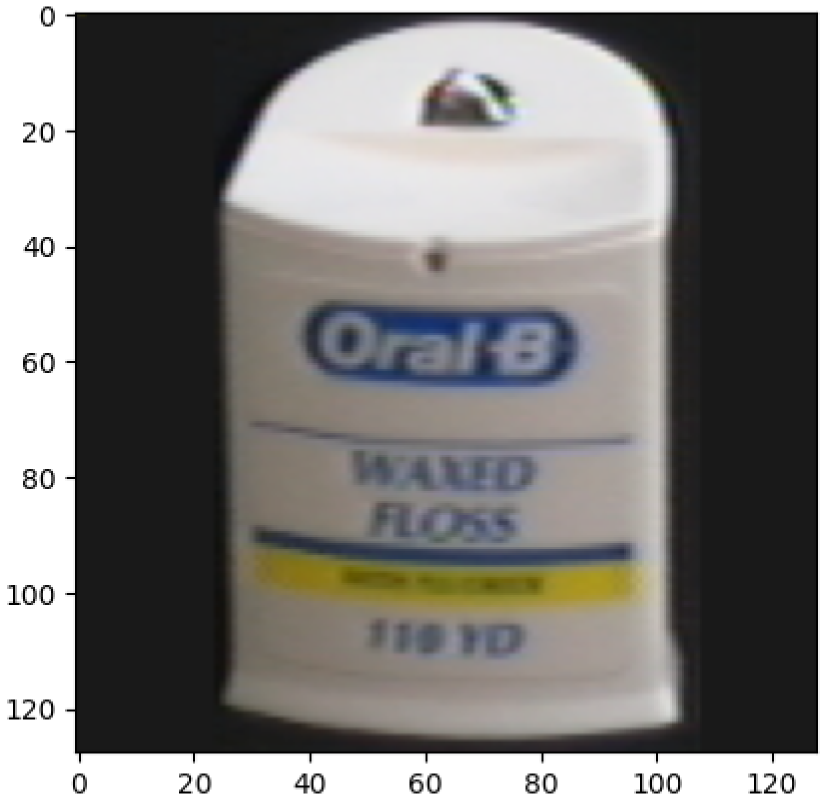
```
#First method PCA:
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
from sklearn.manifold import MDS
from sklearn import datasets
from mnist import MNIST
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
import csv
from matplotlib import cm
```

In [ ]:

```
#Path to images:
path_to_coil = "./coil-100/coil-100/"
```

In [ ]:

```
def show_image(obj_id=9, view_id=5):
    image = Image.open(path_to_coil + f"obj{obj_id}__{view_id}.png")
    plt.imshow(image)

show_image()
```

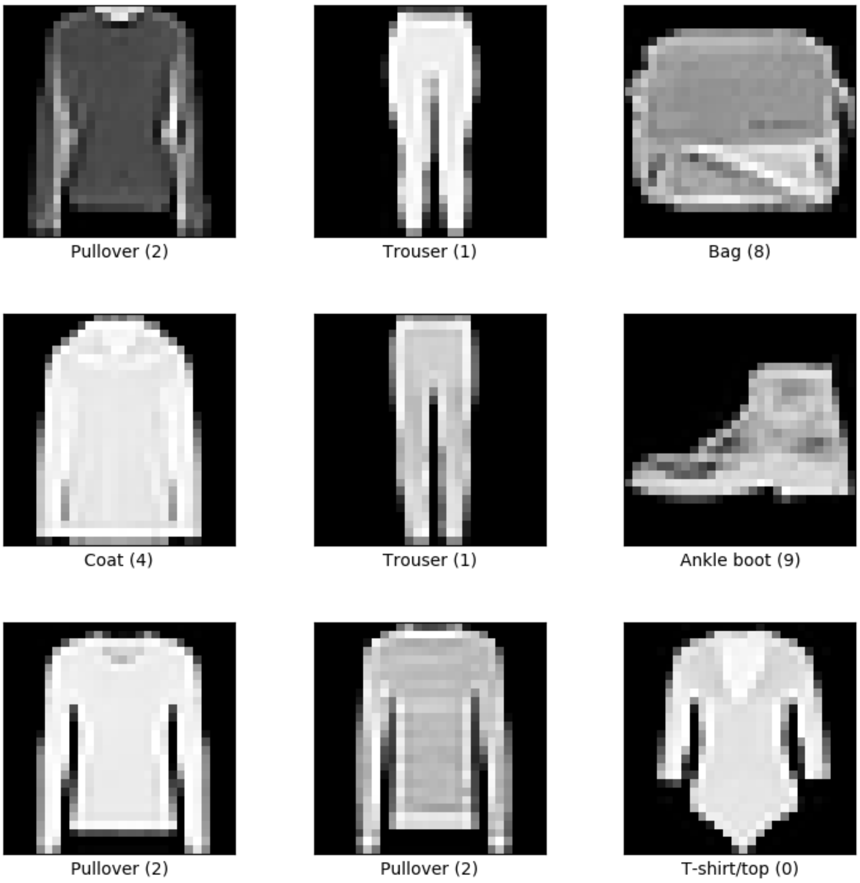


Digits dataset:

It has 10 classes and 180 samples per class, it is available directly through sklearn. Contains handwritten digits.

Fashion-MNIST:

It has 10 classes of types of fashion and gray-scaled pictures of these types of fashion.



Pullover (2) Trouser (1) Bag (8)

Coat (4) Trouser (1) Ankle boot (9)

Pullover (2) Pullover (2) T-shirt/top (0)

COIL-Dataset:

It has 100 different objects, each of this object has 71 pictures with a different perspective.

```
In [ ]: def load_num_of_dataset_digits(number_of_objects = 150):
            X, y = datasets.load_digits(return_X_y=True)
            return X[0:number_of_objects], y[0:number_of_objects]

        def load_num_of_dataset_fashion(number_of_objects = 10):
            mndata = MNIST('fashion')
            images, labels = mndata.load_training()
            return np.asarray(images)[0:number_of_objects], np.asarray(labels)[0:



        def load_num_of_dataset_coil(number_of_objects = 10):
            images = np.array([])
            labels = np.array([])

            obj_images = []
            obj_labels = []
            for obj_id in range(1, number_of_objects + 1):
                for view_id in range(0, 360, 5):
                    image = Image.open(path_to_coil + f"obj{obj_id}__{view_id}.pn
                    array = np.asarray(image)
                    #Reshaping the images from 3d (rgb) into 1d:
                    obj_images.append(array.reshape(-1))
                    obj_labels.append(obj_id)

            images = np.array(obj_images)
            labels = np.array(obj_labels)

            return (images, labels)

        def show_scatter(number_of_classes, images_transformed, labels):
```

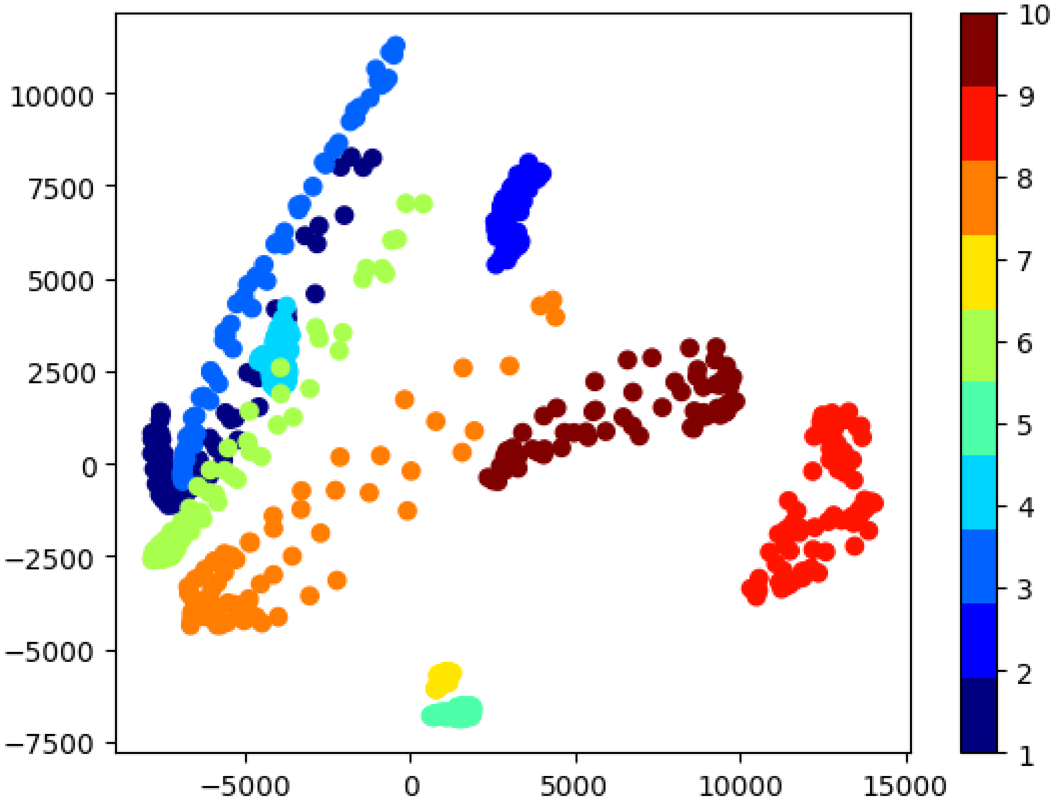
```
    c_map = plt.cm.get_cmap('jet', number_of_classes)

    plt.scatter(images_transformed[:,0], images_transformed[:,1],
                cmap= c_map, c=labels)
    plt.colorbar()
    plt.show()
```

Method PCA for all datasets:

COIL:

In [ ]:
```
number_of_classes = 10
images, labels = load_num_of_dataset_coil(number_of_classes)
#do PCA stuff:
pca = PCA(2)
images_transformed = pca.fit_transform(images)
print(images.shape)
print(images_transformed.shape)
#show scatter plot:
show_scatter(number_of_classes, images_transformed, labels)
```

```
(720, 49152)
(720, 2)
```

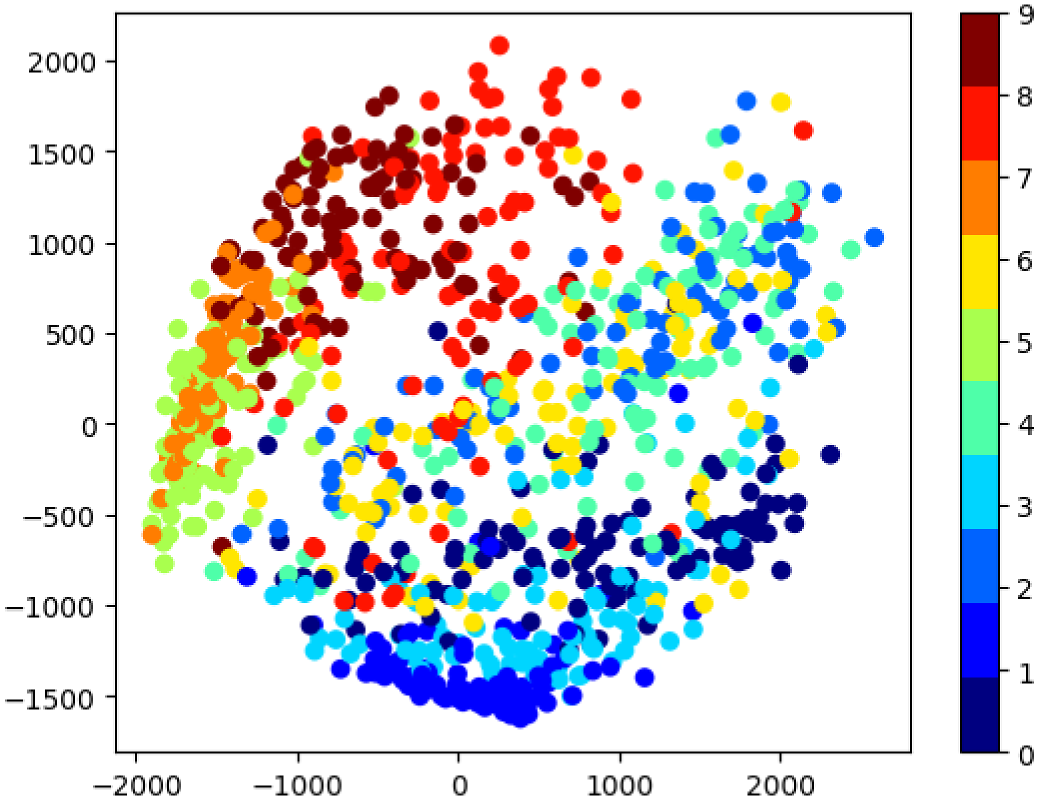


Fashion_MNIST:

In [ ]:
```
number_of_classes = 1000
images, labels = load_num_of_dataset_fashion(number_of_classes)
pca = PCA(2)
```

```
images_transformed = pca.fit_transform(images)
print(images.shape)
print(images_transformed.shape)

show_scatter(10, images_transformed, labels)
```

```
(1000, 784)
(1000, 2)
```

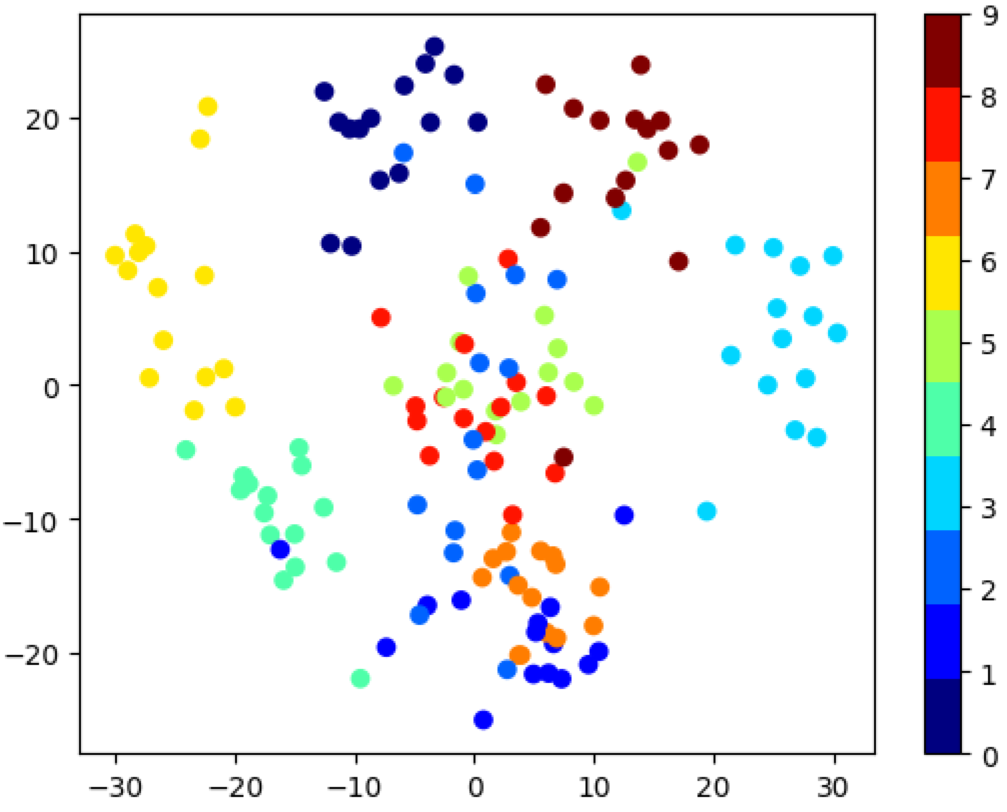


Digits:

In [ ]:

```
X, y = load_num_of_dataset_digits()
pca = PCA(2)
X_transformed = pca.fit_transform(X)
print(X.shape)
print(X_transformed.shape)
show_scatter(10, X_transformed, y)
```

```
(150, 64)
(150, 2)
```

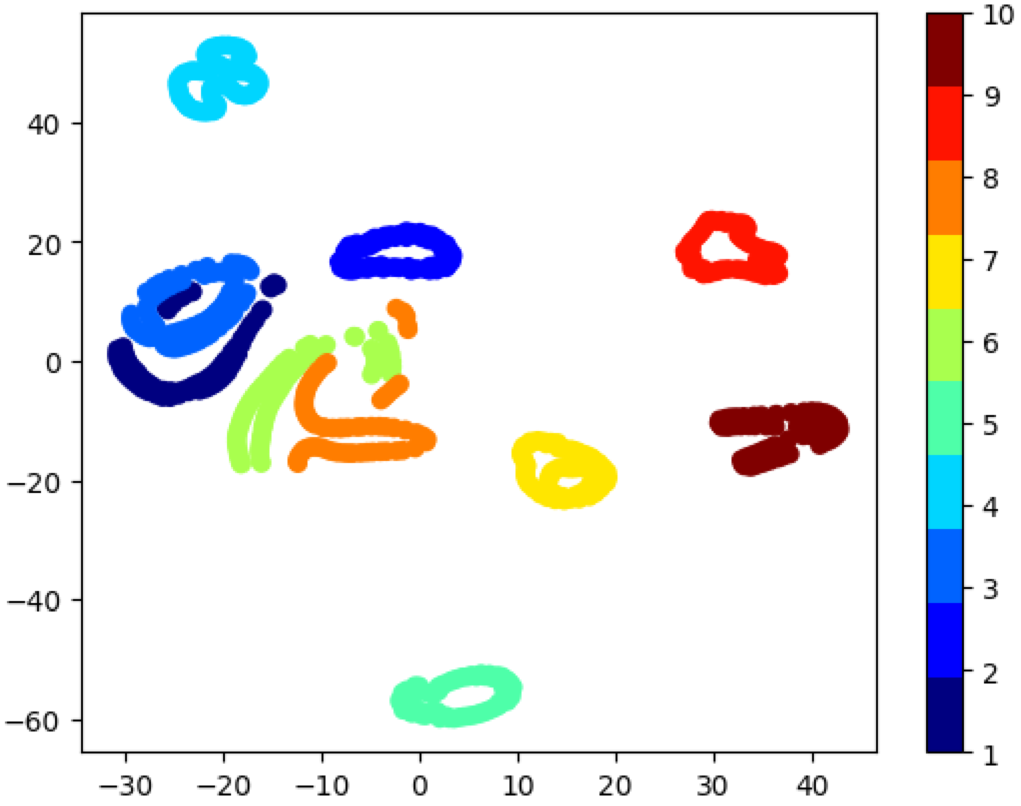


Next method: t-SNE

COIL:

```
In [ ]: number_of_classes = 10
        images, labels = load_num_of_dataset_coil(number_of_classes)
        #do PCA stuff:
        tsne = TSNE(2)
        images_transformed = tsne.fit_transform(images)
        print(images.shape)
        print(images_transformed.shape)
        #show scatter plot:
        show_scatter(number_of_classes, images_transformed, labels)
```

```
(720, 49152)
(720, 2)
```

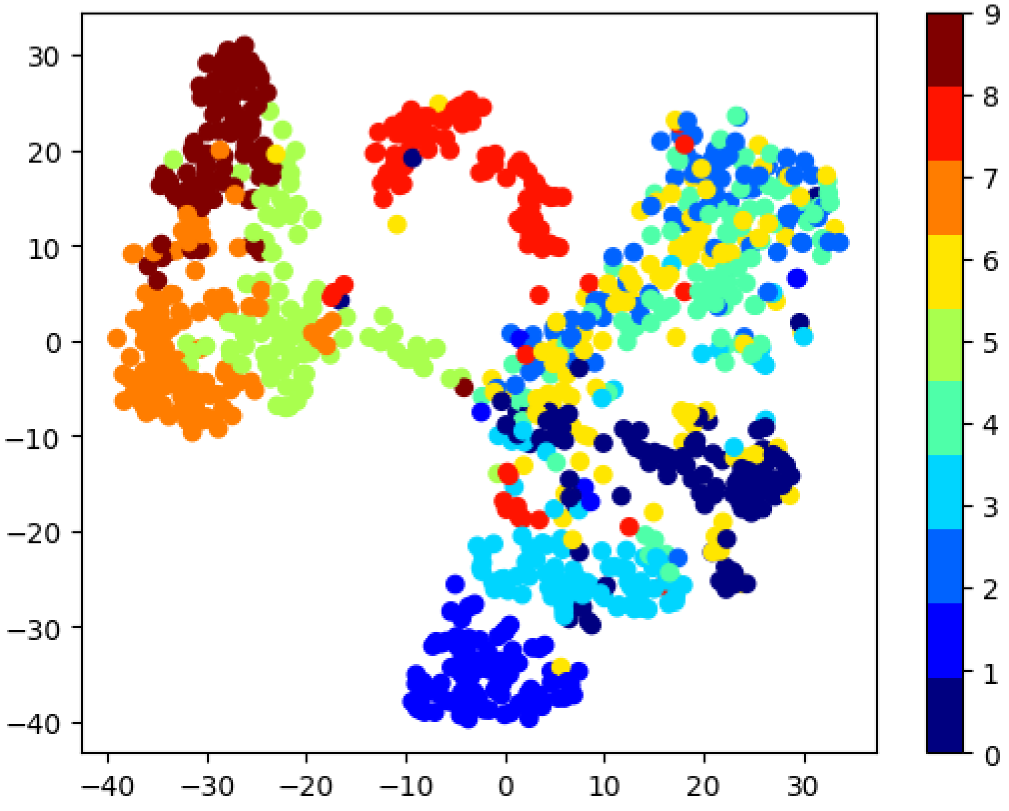


Fashion-MNIST:

In [ ]:

```
number_of_classes = 1000
images, labels = load_num_of_dataset_fashion(number_of_classes)
tsne = TSNE(2)
images_transformed = tsne.fit_transform(images)
print(images.shape)
print(images_transformed.shape)

show_scatter(10, images_transformed, labels)
```

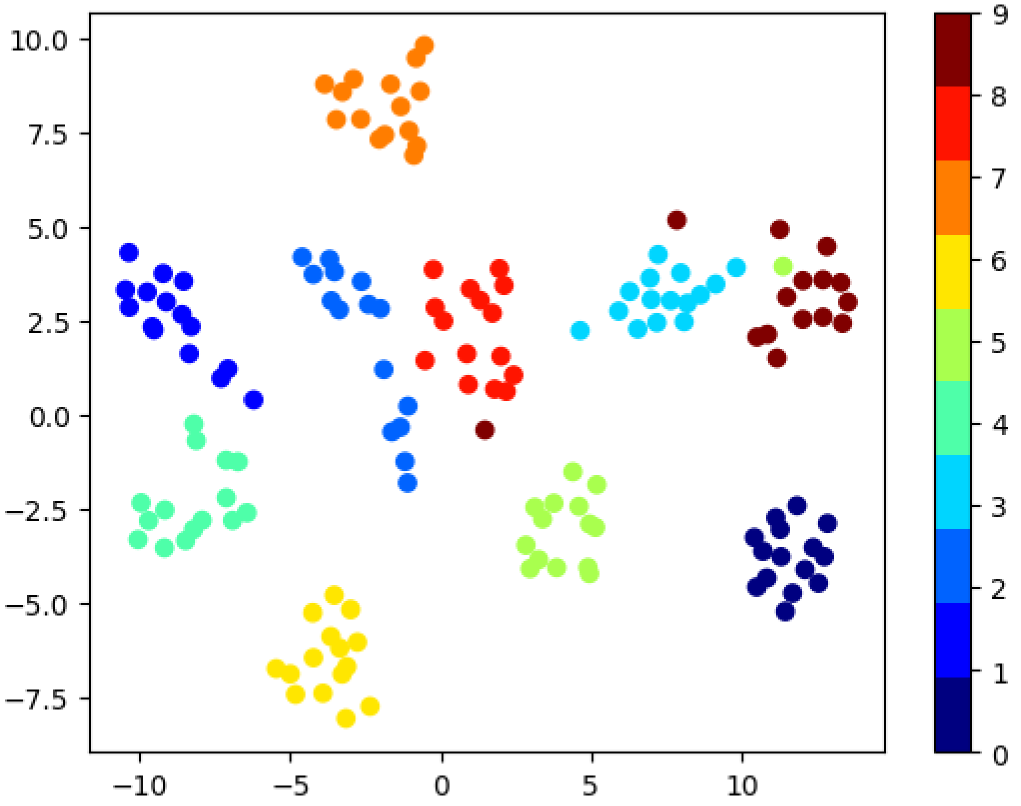
```
(1000, 784)
(1000, 2)
```



Digits:

```
In [ ]: X, y = load_num_of_dataset_digits()
        tsne = TSNE(2)
        X_transformed = tsne.fit_transform(X)
        print(X.shape)
        print(X_transformed.shape)
        show_scatter(10, X_transformed, y)
```

```
(150, 64)
(150, 2)
```



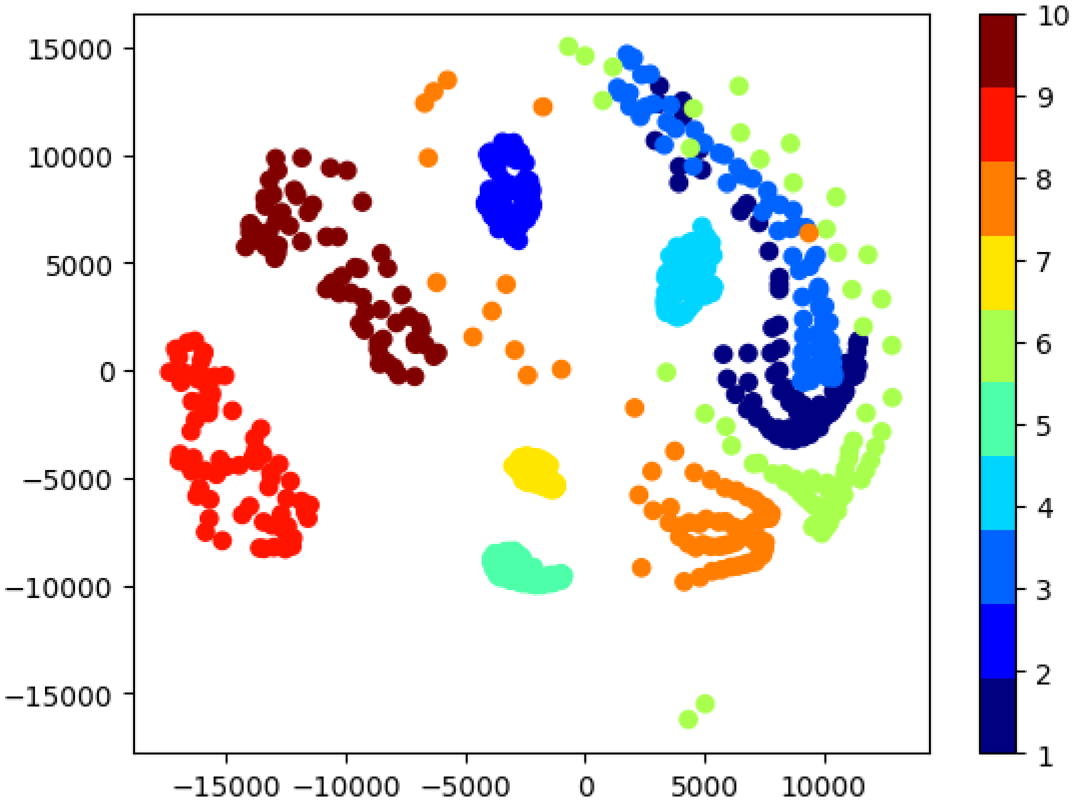
The last method chosen is MDS:

COIL:

In [ ]:

```
number_of_classes = 10
images, labels = load_num_of_dataset_coil(number_of_classes)
#do PCA stuff:
mds = MDS(2)
images_transformed = mds.fit_transform(images)
print(images.shape)
print(images_transformed.shape)
#show scatter plot:
show_scatter(number_of_classes, images_transformed, labels)
```

```
/home/aiko/.local/lib/python3.10/site-packages/sklearn/manifold/_mds.py:
299: FutureWarning: The default value of `normalized_stress` will change
to ``'auto'`` in version 1.4. To suppress this warning, manually set the v
alue of `normalized_stress`.
  warnings.warn(
(720, 49152)
(720, 2)
```

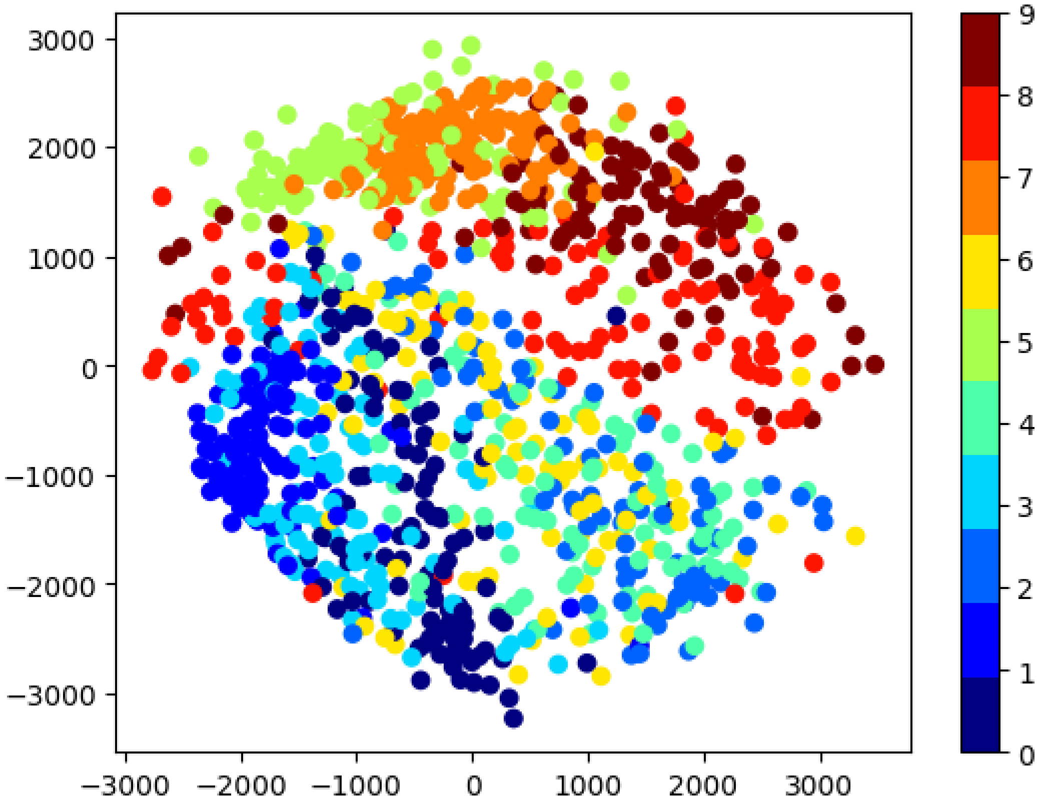


Fashion-MNIST:

In [ ]:

```
number_of_classes = 1000
images, labels = load_num_of_dataset_fashion(number_of_classes)
mds = MDS(2)
images_transformed = mds.fit_transform(images)
print(images.shape)
print(images_transformed.shape)

show_scatter(10, images_transformed, labels)
```

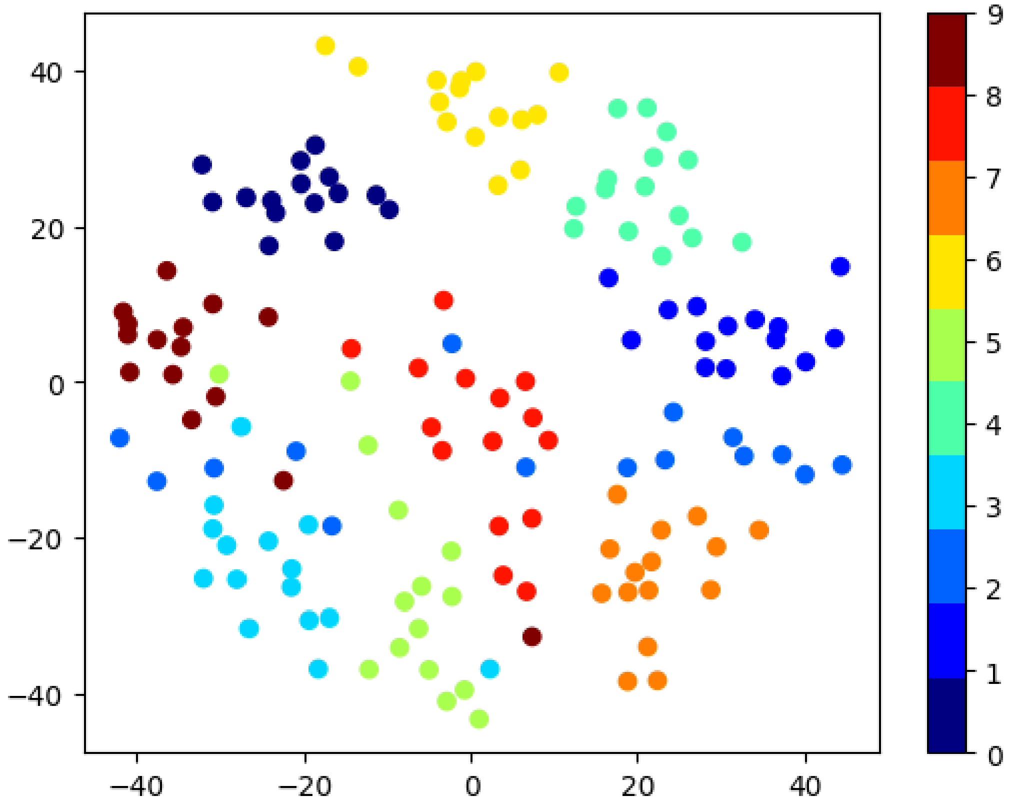
```
/home/aiko/.local/lib/python3.10/site-packages/sklearn/manifold/_mds.py:
299: FutureWarning: The default value of `normalized_stress` will change
to `'auto'` in version 1.4. To suppress this warning, manually set the v
alue of `normalized_stress`.
  warnings.warn(
(1000, 784)
(1000, 2)
```



Digits:

```
In [ ]: X, y = load_num_of_dataset_digits()
        mds = MDS(2)
        X_transformed = mds.fit_transform(X)
        print(X.shape)
        print(X_transformed.shape)
        show_scatter(10, X_transformed, y)
```

```
/home/aiko/.local/lib/python3.10/site-packages/sklearn/manifold/_mds.py:
299: FutureWarning: The default value of `normalized_stress` will change
to `'auto'` in version 1.4. To suppress this warning, manually set the v
alue of `normalized_stress`.
  warnings.warn(
(150, 64)
(150, 2)
```

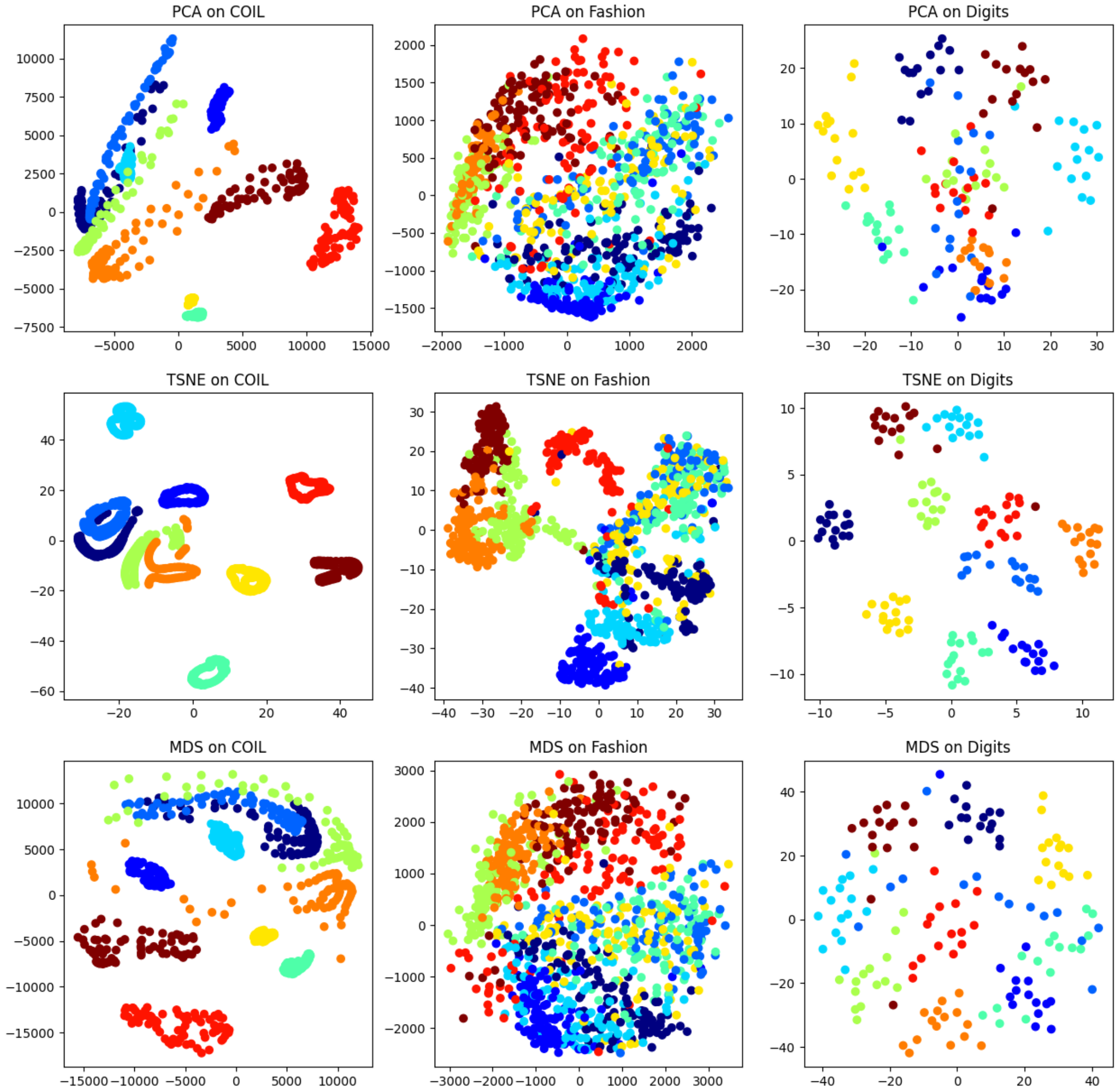


Showing all results, for better comparison:

In [ ]:

```
def complete_routine(X, y, method):
    method_obj = method(2)
    X_transformed = method_obj.fit_transform(X)
    return X_transformed, y

methods = [PCA, TSNE, MDS]
all_datasets = [(load_num_of_dataset_coil, 10, "COIL"),
                (load_num_of_dataset_fashion, 1000, "Fashion"),
                (load_num_of_dataset_digits, 150, "Digits")]

fig, axs = plt.subplots(3, 3, figsize=(15,15))

for i_m, method in enumerate(methods):
    for i_d, (current_dataset, number_of_objects, name) in enumerate(all_
        X, y = complete_routine(*current_dataset(number_of_objects), meth
        # all into one plot:
        c_map = plt.cm.get_cmap('jet', number_of_objects)

        axs[i_m, i_d].scatter(X[:,0], X[:,1],
                    cmap= c_map, c=y)
        axs[i_m, i_d].set_title(f"{method.__name__} on {name}")

fig.show()
```

```
/home/aiko/.local/lib/python3.10/site-packages/sklearn/manifold/_mds.py:
299: FutureWarning: The default value of `normalized_stress` will change
to ``'auto'`` in version 1.4. To suppress this warning, manually set the v
alue of `normalized_stress`.
  warnings.warn(
/home/aiko/.local/lib/python3.10/site-packages/sklearn/manifold/_mds.py:
299: FutureWarning: The default value of `normalized_stress` will change
to ``'auto'`` in version 1.4. To suppress this warning, manually set the v
alue of `normalized_stress`.
  warnings.warn(
/home/aiko/.local/lib/python3.10/site-packages/sklearn/manifold/_mds.py:
299: FutureWarning: The default value of `normalized_stress` will change
to ``'auto'`` in version 1.4. To suppress this warning, manually set the v
alue of `normalized_stress`.
  warnings.warn(
/tmp/ipykernel_45492/2877936415.py:23: UserWarning: Matplotlib is curren
tly using module://matplotlib_inline.backend_inline, which is a non-GUI
backend, so cannot show the figure.
  fig.show()
```

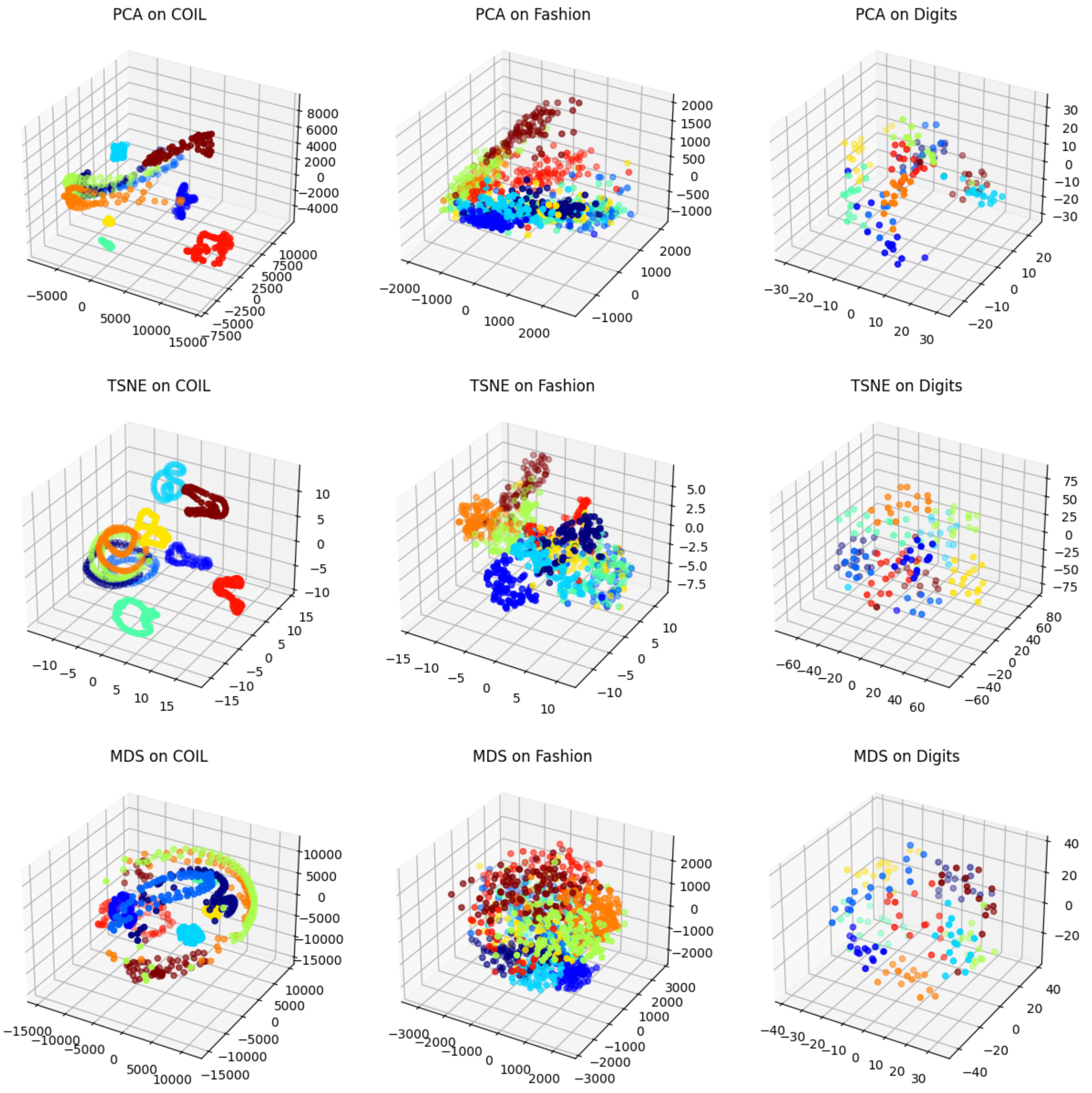


(may be interesting) 3D instead of 2D plots:

In [ ]:

```
def complete_routine3d(X, y, method):
    method_obj = method(3)
    X_transformed = method_obj.fit_transform(X)
    return X_transformed, y

fig, axs = plt.subplots(3, 3, figsize=(15,15), subplot_kw=dict(projection

for i_m, method in enumerate(methods):
    for i_d, (current_dataset, number_of_objects, name) in enumerate(all_
        X, y = complete_routine3d(*current_dataset(number_of_objects), me
        # all into one plot:
        c_map = plt.cm.get_cmap('jet', number_of_objects)

        axs[i_m, i_d].scatter(X[:,0], X[:,1], X[:,2],
                    cmap= c_map, c=y)
        axs[i_m, i_d].set_title(f"{method.__name__} on {name}")

fig.show()
```

```
/home/aiko/.local/lib/python3.10/site-packages/sklearn/manifold/_mds.py:
299: FutureWarning: The default value of `normalized_stress` will change
to `'auto'` in version 1.4. To suppress this warning, manually set the v
alue of `normalized_stress`.
  warnings.warn(
/home/aiko/.local/lib/python3.10/site-packages/sklearn/manifold/_mds.py:
299: FutureWarning: The default value of `normalized_stress` will change
to `'auto'` in version 1.4. To suppress this warning, manually set the v
alue of `normalized_stress`.
  warnings.warn(
/home/aiko/.local/lib/python3.10/site-packages/sklearn/manifold/_mds.py:
299: FutureWarning: The default value of `normalized_stress` will change
to `'auto'` in version 1.4. To suppress this warning, manually set the v
alue of `normalized_stress`.
  warnings.warn(
/tmp/ipykernel_45492/3849120921.py:18: UserWarning: Matplotlib is curren
tly using module://matplotlib_inline.backend_inline, which is a non-GUI
backend, so cannot show the figure.
  fig.show()
```



PCA on COIL

PCA on Fashion

PCA on Digits

TSNE on COIL

TSNE on Fashion

TSNE on Digits

MDS on COIL

MDS on Fashion

MDS on Digits

I'm not quite sure how to numericaly evaluate the results.

In [ ]: