In [ ]:

```
import numpy as np
from PIL import Image
import copy
```

In [ ]:

```
# constants
CLASS = 200
SAMPLES_PER_CLASS = 500
TOTAL_SAMPLES = CLASS * SAMPLES_PER_CLASS
COLOR_CHANNELS = 3
IMAGE_WIDTH = 64
IMAGE_HEIGHT = 64
VAL_SAMPLES = 10000
```

In [ ]:

```
# read all of the word net id
wnids = [id.strip('\n') for id in open('/Users/kimjaspermui/Desktop/data/tiny-imagenet-200/wnids.txt').readlines()]

# data will store all of the data
data = {}

# train data
data['train'] = {}
data['train']['data'] = np.ndarray(shape=(TOTAL_SAMPLES, COLOR_CHANNELS * IMAGE_WIDTH * IMAGE_HEIGHT ), dtype=np.uint8)
data['train']['target'] = np.ndarray(shape=(TOTAL_SAMPLES,), dtype=np.uint8)

# validation data
data['val'] = {}
data['val']['data'] = np.ndarray(shape=(VAL_SAMPLES, COLOR_CHANNELS * IMAGE_WIDTH * IMAGE_HEIGHT), dtype=np.uint8)
data['val']['target'] = np.ndarray(shape=(VAL_SAMPLES,), dtype=np.uint8)

# iterate through work net ids
print("storing training:")
for i in range(len(wnids)):
    wnid = wnids[i]
    print("%s: %d / %d" % (wnid, i + 1, len(wnids)))
    for j in range(500):
        temp = []
        path = "/Users/kimjaspermui/Desktop/data/tiny-imagenet-200/train/{0}/images/{0}_{1}.JPEG".format(wnid, j)
        test = np.array(Image.open(path).convert('RGB')).transpose(2,0,1)
        for k in range(len(test)):
            temp.append(test[k].flatten())
        data['train']['data'][i * SAMPLES_PER_CLASS + j] = np.array(temp).flatten()
        data['train']['target'][i * SAMPLES_PER_CLASS + j] = i

# get the validation data
print("storing validation:")
for i, line in enumerate(map(lambda s: s.strip(), open('/Users/kimjaspermui/Desktop/data/tiny-imagenet-200/val/val_annotations.txt'))):
    name, wnid = line.split('\t')[0:2]
    temp = []
    path = "/Users/kimjaspermui/Desktop/data/tiny-imagenet-200/val/images/{0}".format(name)
    test = np.array(Image.open(path).convert('RGB')).transpose(2,0,1)
    for k in range(len(test)):
        temp.append(test[k].flatten())
    data['val']['data'][i] = np.array(temp).flatten()
    data['val']['target'][i] = wnids.index(wnid)
```

In [ ]:

```
import cPickle
import gzip

def save_zipped_pickle(obj, filename, protocol=-1):
    with gzip.open(filename, 'wb') as f:
        cPickle.dump(obj, f, protocol)

def load_zipped_pickle(filename):
    with gzip.open(filename, 'rb') as f:
        loaded_object = cPickle.load(f)
        return loaded_object
```

In [ ]:

```
print("storing to file...")
save_zipped_pickle(data, "tinyImageData")
```

In [ ]:

```
print("getting the data from file...")
tempData = load_zipped_pickle("tinyImageData")
```