

Problem

AI Track's Exercise to create a basic CNN to recognize different animals from the Chinese Zodiac

The Road Ahead

We break the notebook into separate steps. Feel free to use the links below to navigate the notebook.

- [Step 0](#): Import Datasets
- [Step 1](#): Visualize a Batch of Training Data
- [Step 2](#): Create a CNN to recognize Chinese Zodiac (from Scratch)
- [Step 3](#): Create a CNN to recognize Chinese Zodiac (using Transfer Learning)
- [Step 4](#): Write Algorithm
- [Step 5](#): Test Algorithm

In [0]:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import os
import torch
from torchvision import datasets
import torchvision.transforms as transforms
from torchvision.datasets import ImageFolder
from torch.utils.data import DataLoader
from PIL import ImageFile
from PIL import Image
ImageFile.LOAD_TRUNCATED_IMAGES = True
from torch import optim
```

In [0]:

```
%ls
```

In [10]:

```
# check if CUDA is available
train_on_gpu = torch.cuda.is_available()

if not train_on_gpu:
    print('CUDA is not available.  Training on CPU ...')
else:
    print('CUDA is available!  Training on GPU ...')
```

CUDA is available! Training on GPU ...

Step 0: Import Datasets

Since the data already hosted in Kaggel, we can directly add data on the top right corner of this notebook.

In [11]:

```
# Mount Google Drive
from google.colab import drive

ROOT = "/content/drive"      # default for the drive
print(ROOT)

drive.mount(ROOT, force_remount=True)      # we mount the drive at /content/drive
```

/content/drive

Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aob&response_type=code&scope=email%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdocs.test%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.readonly%20https%3a%2f%2fwww.googleapis.com%2fauth%2fpeopleapi.readonly

Enter your authorization code:

.....

Mounted at /content/drive

In [12]:

```
%cd /content/drive/My\ Drive/MyDrive/Udacity/CNN\ Chinese\ Zodiac\ Signs

/content/drive/My Drive/MyDrive/Udacity/CNN Chinese Zodiac Signs
```

In [0]:

```
# uncomment if you want to create directory checkpoint, best_model
# %mkdir checkpoint best_model
```

In [5]:

```
%ls
```

```
best_model/          Network_architectures_1.png
checkpoint/          Network_architectures_2.png
'Chinese_Zodiac_Signs (1).gdoc'  Network_architectures.png
Chinese_Zodiac_Signs.gdoc      serialized_model/
'Chinese_Zodiac_Signs.ipynb'    test_images/
input/
```

In [0]:

```
#!/rm ./input/chinese-zodiac-signs/signs/valid/tiger/00000576.jpg ./input/chinese-zodiac-signs/signs/valid/snake/serpant0015.jpg ./input/chinese-zodiac-signs/signs/valid/snake/00000576.jpg ./input/chinese-zodiac-signs/signs/train/rabbit/00000134.jpg ./input/chinese-zodiac-signs/signs/train/goat/00000582.jpg ./input/chinese-zodiac-signs/signs/train/monkey/00000585.jpg ./input/chinese-zodiac-signs/signs/train/snake/00000337.jpg ./input/chinese-zodiac-signs/signs/train/snake/00000116.jpg ./input/chinese-zodiac-signs/signs/train/rooster/00000216.jpg ./input/chinese-zodiac-signs/signs/train/pig/49476.html ./input/chinese-zodiac-signs/signs/train/pig/00000444.jpg
```

In [0]:

```
# number of subprocesses to use for data loading
num_workers = 0
# how many samples per batch to load
batch_size = 20

image_transformation = {
    # Train uses data augmentation
    'train':
        transforms.Compose([
            transforms.RandomResizedCrop(size=256, scale=(0.8, 1.0)),
            transforms.RandomRotation(degrees=15),
            transforms.ColorJitter(),
            transforms.RandomHorizontalFlip(),
            transforms.CenterCrop(size=224), # Image net standards
            transforms.ToTensor(),
            transforms.Normalize([0.485, 0.456, 0.406],
                                [0.229, 0.224, 0.225]) # Imagenet standards
        ]),
    # Validation does not use augmentation
    'valid':
        transforms.Compose([
            transforms.Resize(size=256),
            transforms.CenterCrop(size=224),
            transforms.ToTensor(),
            transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
        ]),
    # Test does not use augmentation
    'test':
        transforms.Compose([
            transforms.Resize(size=256),
            transforms.CenterCrop(size=224),
            transforms.ToTensor(),
            transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
        ]),
}

# choose the training and test datasets
data_dir = './input/chinese-zodiac-signs'
train_dir = data_dir + '/train'
valid_dir = data_dir + '/valid'
test_dir = data_dir + '/test'

image_datasets = {
    'train' : ImageFolder(root=train_dir,transform=image_transformation['train']
    ),
    'valid' : ImageFolder(root=valid_dir,transform=image_transformation['valid']
    ),
    'test' : ImageFolder(root=test_dir,transform=image_transformation['test']),
}

data_loaders = {
    'train' : DataLoader(image_datasets['train'],batch_size = batch_size,shuffle
    =True),
    'valid' : DataLoader(image_datasets['valid'],batch_size = batch_size,shuffle
    =True),
    'test' : DataLoader(image_datasets['test'],batch_size = batch_size,shuffle=
    True),
}
```


In [0]:

```

# Empty lists
"""categories = []
img_train_categories = []
img_valid_categories = []
img_test_categories = []
n_train = []
n_valid = []
n_test = []
hs = []
ws = []

# Iterate through each category
for d in os.listdir(train_dir):
    categories.append(d)

    # Number of each image
    train_imgs = os.listdir(train_dir + '/' + d)
    valid_imgs = os.listdir(valid_dir + '/' + d)
    test_imgs = os.listdir(test_dir + '/' + d)
    n_train.append(len(train_imgs))
    n_valid.append(len(valid_imgs))
    n_test.append(len(test_imgs))

    # Find stats for train images
    for i in train_imgs:
        img_train_categories.append(d)
        img_file = train_dir + '/' + d + '/' + i
        if(img_file != './input/chinese-zodiac-signs/signs/train/rabbit/0000013
4.jpg'
            and img_file != './input/chinese-zodiac-signs/signs/train/goat/0000058
2.jpg'
            and img_file != './input/chinese-zodiac-signs/signs/train/monkey/00000
585.jpg'
            and img_file != './input/chinese-zodiac-signs/signs/train/snake/0000003
37.jpg'
            and img_file != './input/chinese-zodiac-signs/signs/train/snake/0000001
16.jpg'
            and img_file != './input/chinese-zodiac-signs/signs/train/rooster/0000
0216.jpg'
            and img_file != './input/chinese-zodiac-signs/signs/train/pig/49476.ht
ml'
            and img_file != './input/chinese-zodiac-signs/signs/train/pig/0000044
4.jpg'):
            img = Image.open(img_file)
            img_array = np.array(img)
            # Shape
            hs.append(img_array.shape[0])
            ws.append(img_array.shape[1])

    # Find stats for validation images
    for i in valid_imgs:
        img_valid_categories.append(d)
        img_file = valid_dir + '/' + d + '/' + i
        if(img_file != './input/chinese-zodiac-signs/signs/valid/snake/serpant00
15.jpg'
            and img_file != './input/chinese-zodiac-signs/signs/valid/snake/0000005
76.jpg'
            and img_file != './input/chinese-zodiac-signs/signs/valid/tiger/0000005

```

```

76.jpg'
    and img_file != './input/chinese-zodiac-signs/signs/valid/dragon/00000
415.jpg'):
    img = Image.open(img_file)
    img_array = np.array(img)
    # Shape
    hs.append(img_array.shape[0])
    ws.append(img_array.shape[1])

    # Find stats for validation images
    for i in test_imgs:
        img_test_categories.append(d)
        img_file = test_dir + '/' + d + '/' + i
        if(img_file != './input/chinese-zodiac-signs/signs/test/monkey/00000236.
jpeg'):
            img = Image.open(img_file)
            img_array = np.array(img)
            # Shape
            hs.append(img_array.shape[0])
            ws.append(img_array.shape[1])

# Dataframe of categories
cat_df = pd.DataFrame({'category': categories,
                        'n_train': n_train,
                        'n_valid': n_valid,
                        'n_test': n_test}).\
    sort_values('category')

# Dataframe of training images
a = {'train category': img_train_categories,
     'height': hs,
     'width': ws}
image_df = pd.DataFrame.from_dict(a, orient='index')
image_df.transpose()

cat_df.sort_values('n_train', ascending=False, inplace=True)
cat_df.head()
cat_df.tail() """

```

Out[0]:

	category	n_train	n_valid	n_test
9	rooster	599	54	55
4	goat	598	55	55
0	pig	598	54	55
8	rabbit	598	55	55
11	snake	598	52	55

In [0]:

```

#%cd '/kaggle/input/chinese-zodiac-signs/signs/train/rabbit/'

```

In [0]:

```
#im = Image.open("/kaggle/input/chinese-zodiac-signs/signs/train/goat/00000582.jpg")
#im.verify()
```

Step 1: Visualize a Batch of Training Data

In [0]:

```
import matplotlib.pyplot as plt
%matplotlib inline

# helper function to un-normalize and display an image
def imshow(img):
    img = img / 2 + 0.5 # unnormalize
    plt.imshow(np.transpose(img, (1, 2, 0))) # convert from Tensor image
```

In [19]:

```
class_names = [item[0:].replace("_", " ") for item in image_datasets['train'].classes]
class_names
```

Out[19]:

```
['dog',
 'dragon',
 'goat',
 'horse',
 'monkey',
 'ox',
 'pig',
 'rabbit',
 'ratt',
 'rooster',
 'snake',
 'tiger']
```

In [0]:

```
# obtain one batch of training images
dataiter = iter(data_loaders['train'])
images, labels = dataiter.next()
images = images.numpy() # convert images to numpy for display

# plot the images in the batch, along with the corresponding labels
fig = plt.figure(figsize=(25, 4))
# display 20 images
for idx in np.arange(20):
    ax = fig.add_subplot(2, 20/2, idx+1, xticks=[], yticks=[])
    imshow(images[idx])
    ax.set_title(class_names[labels[idx]])
```


[illegible]

Step 3: Create a CNN to Classify Dog Breeds (using Transfer Learning)

In [14]:

```
# check if CUDA is available
USE_GPU = True # currently GPU not available
use_cuda = USE_GPU and torch.cuda.is_available()
print(use_cuda)
```

True

In [0]:

```
import torchvision.models as models
import torch.nn as nn
```

Model Mobilev2

In [91]:

```
## Specify model architecture  
model_transfer_mobilev2 = models.mobilenet_v2(pretrained=True)  
  
print(model_transfer_mobilev2)
```

```

MobileNetV2(
  (features): Sequential(
    (0): ConvBNReLU(
      (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(2, 2), padding=
(1, 1), bias=False)
      (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, tra
ck_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (1): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), pad
ding=(1, 1), groups=32, bias=False)
          (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2d(32, 16, kernel_size=(1, 1), stride=(1, 1), bias=
False)
        (2): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
      )
    )
    (2): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(16, 96, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(96, 96, kernel_size=(3, 3), stride=(2, 2), pad
ding=(1, 1), groups=96, bias=False)
          (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(96, 24, kernel_size=(1, 1), stride=(1, 1), bias=
False)
        (3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
      )
    )
    (3): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(24, 144, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(144, 144, kernel_size=(3, 3), stride=(1, 1), p
adding=(1, 1), groups=144, bias=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)

```

```

    )
    (2): Conv2d(144, 24, kernel_size=(1, 1), stride=(1, 1), bias
=False)
    (3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
    )
    )
    (4): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(24, 144, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(144, 144, kernel_size=(3, 3), stride=(2, 2), p
adding=(1, 1), groups=144, bias=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(144, 32, kernel_size=(1, 1), stride=(1, 1), bias
=False)
        (3): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
      )
    )
    (5): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(192, 192, kernel_size=(3, 3), stride=(1, 1), p
adding=(1, 1), groups=192, bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(192, 32, kernel_size=(1, 1), stride=(1, 1), bias
=False)
        (3): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
      )
    )
    (6): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(

```

```

        (0): Conv2d(192, 192, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=192, bias=False)
        (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU6(inplace=True)
    )
    (2): Conv2d(192, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    )
    (7): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(192, 192, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=192, bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(192, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (8): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=384, bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (9): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)

```

```

        (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU6(inplace=True)
    )
    (1): ConvBNReLU(
        (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=384, bias=False)
        (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU6(inplace=True)
    )
    (2): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    )
    (10): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=384, bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (11): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=384, bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(384, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
  )

```

```

(12): InvertedResidual(
  (conv): Sequential(
    (0): ConvBNReLU(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (1): ConvBNReLU(
      (0): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=576, bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (2): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(13): InvertedResidual(
  (conv): Sequential(
    (0): ConvBNReLU(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (1): ConvBNReLU(
      (0): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=576, bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (2): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(14): InvertedResidual(
  (conv): Sequential(
    (0): ConvBNReLU(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (1): ConvBNReLU(
      (0): Conv2d(576, 576, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=576, bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (2): Conv2d(576, 160, kernel_size=(1, 1), stride=(1, 1), bias=False)
  )
)

```



```

s=False)
    (3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    )
    )
    (15): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), b
ias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), p
adding=(1, 1), groups=960, bias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
        (3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
    )
    (16): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), b
ias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), p
adding=(1, 1), groups=960, bias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
        (3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
    )
    (17): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), b
ias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), p
adding=(1, 1), groups=960, bias=False)

```

```

        (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (2): ReLU6(inplace=True)
    )
    (2): Conv2d(960, 320, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): BatchNorm2d(320, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    )
    )
    (18): ConvBNReLU(
    (0): Conv2d(320, 1280, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(1280, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU6(inplace=True)
    )
    )
    (classifier): Sequential(
    (0): Dropout(p=0.2, inplace=False)
    (1): Linear(in_features=1280, out_features=1000, bias=True)
    )
    )

```

In [93]:

```
pip install torchsummary
```

Requirement already satisfied: torchsummary in /usr/local/lib/python3.6/dist-packages (1.5.1)

In [94]:

```

# Number of gpus
if train_on_gpu:
    gpu_count = torch.cuda.device_count()
    print(f'{gpu_count} gpus detected.')
    if gpu_count > 1:
        multi_gpu = True
    else:
        multi_gpu = False

```

1 gpus detected.

In [0]:

```

# Freeze parameters so we don't backprop through them
for param in model_transfer_mobilev2.parameters():
    param.requires_grad = False

n_inputs = model_transfer_mobilev2.classifier[1].in_features
n_classes = len(class_names)
model_transfer_mobilev2.classifier = nn.Sequential(
    nn.Dropout(0.2, inplace = False),
    nn.Linear(n_inputs, n_classes, bias = True))

# Unfreeze training for classifier layers
for param in model_transfer_mobilev2.classifier.parameters():
    param.requires_grad = True

```

In [0]:

```
if use_cuda:  
    model_transfer_mobilev2 = model_transfer_mobilev2.cuda()
```

In [97]:

```
print(model_transfer_mobilev2)
```

```

MobileNetV2(
  (features): Sequential(
    (0): ConvBNReLU(
      (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(2, 2), padding=
(1, 1), bias=False)
      (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, tra
ck_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (1): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), pad
ding=(1, 1), groups=32, bias=False)
          (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2d(32, 16, kernel_size=(1, 1), stride=(1, 1), bias=
False)
        (2): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
      )
    )
    (2): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(16, 96, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(96, 96, kernel_size=(3, 3), stride=(2, 2), pad
ding=(1, 1), groups=96, bias=False)
          (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(96, 24, kernel_size=(1, 1), stride=(1, 1), bias=
False)
        (3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
      )
    )
    (3): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(24, 144, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(144, 144, kernel_size=(3, 3), stride=(1, 1), p
adding=(1, 1), groups=144, bias=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)

```

```

    )
    (2): Conv2d(144, 24, kernel_size=(1, 1), stride=(1, 1), bias
=False)
    (3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
    )
    )
    (4): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(24, 144, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(144, 144, kernel_size=(3, 3), stride=(2, 2), p
adding=(1, 1), groups=144, bias=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(144, 32, kernel_size=(1, 1), stride=(1, 1), bias
=False)
        (3): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
      )
    )
    (5): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(192, 192, kernel_size=(3, 3), stride=(1, 1), p
adding=(1, 1), groups=192, bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(192, 32, kernel_size=(1, 1), stride=(1, 1), bias
=False)
        (3): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
      )
    )
    (6): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(

```

```

        (0): Conv2d(192, 192, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=192, bias=False)
        (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU6(inplace=True)
    )
    (2): Conv2d(192, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    )
    (7): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(192, 192, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=192, bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(192, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (8): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=384, bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    )
    (9): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)

```

```

        (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU6(inplace=True)
    )
    (1): ConvBNReLU(
        (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=384, bias=False)
        (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU6(inplace=True)
    )
    (2): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    )
    (10): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=384, bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (11): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=384, bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(384, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
  )

```



```

(12): InvertedResidual(
  (conv): Sequential(
    (0): ConvBNReLU(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (1): ConvBNReLU(
      (0): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=576, bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (2): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(13): InvertedResidual(
  (conv): Sequential(
    (0): ConvBNReLU(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (1): ConvBNReLU(
      (0): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=576, bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (2): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(14): InvertedResidual(
  (conv): Sequential(
    (0): ConvBNReLU(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (1): ConvBNReLU(
      (0): Conv2d(576, 576, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=576, bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (2): Conv2d(576, 160, kernel_size=(1, 1), stride=(1, 1), bias=False)
  )
)

```

```

s=False)
    (3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    )
    )
    (15): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), b
ias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), p
adding=(1, 1), groups=960, bias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
        (3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
    )
    (16): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), b
ias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), p
adding=(1, 1), groups=960, bias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
        (3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
    )
    (17): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), b
ias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), p
adding=(1, 1), groups=960, bias=False)

```

```
        (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU6(inplace=True)
    )
    (2): Conv2d(960, 320, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): BatchNorm2d(320, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    )
    (18): ConvBNReLU(
      (0): Conv2d(320, 1280, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(1280, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
  )
  (classifier): Sequential(
    (0): Dropout(p=0.2, inplace=False)
    (1): Linear(in_features=1280, out_features=12, bias=True)
  )
)
```

In [98]:

```
# Useful for examining network
from torchsummary import summary
if USE_GPU and use_cuda:
    if multi_gpu:
        summary(
            model_transfer_mobilev2.module,
            input_size=(3, 224, 224),
            batch_size=batch_size,
            device='cuda')
    else:
        summary(
            model_transfer_mobilev2, input_size=(3, 224, 224), batch_size=batch_
size, device='cuda')
else:
    summary(
        model_transfer_mobilev2, input_size=(3, 224, 224), batch_size=batch_
size, device='cpu')
```

Layer (type)	Output Shape	Param #
Conv2d-1	[20, 32, 112, 112]	864
BatchNorm2d-2	[20, 32, 112, 112]	64
ReLU6-3	[20, 32, 112, 112]	0
Conv2d-4	[20, 32, 112, 112]	288
BatchNorm2d-5	[20, 32, 112, 112]	64
ReLU6-6	[20, 32, 112, 112]	0
Conv2d-7	[20, 16, 112, 112]	512
BatchNorm2d-8	[20, 16, 112, 112]	32
InvertedResidual-9	[20, 16, 112, 112]	0
Conv2d-10	[20, 96, 112, 112]	1,536
BatchNorm2d-11	[20, 96, 112, 112]	192
ReLU6-12	[20, 96, 112, 112]	0
Conv2d-13	[20, 96, 56, 56]	864
BatchNorm2d-14	[20, 96, 56, 56]	192
ReLU6-15	[20, 96, 56, 56]	0
Conv2d-16	[20, 24, 56, 56]	2,304
BatchNorm2d-17	[20, 24, 56, 56]	48
InvertedResidual-18	[20, 24, 56, 56]	0
Conv2d-19	[20, 144, 56, 56]	3,456
BatchNorm2d-20	[20, 144, 56, 56]	288
ReLU6-21	[20, 144, 56, 56]	0
Conv2d-22	[20, 144, 56, 56]	1,296
BatchNorm2d-23	[20, 144, 56, 56]	288
ReLU6-24	[20, 144, 56, 56]	0
Conv2d-25	[20, 24, 56, 56]	3,456
BatchNorm2d-26	[20, 24, 56, 56]	48
InvertedResidual-27	[20, 24, 56, 56]	0
Conv2d-28	[20, 144, 56, 56]	3,456
BatchNorm2d-29	[20, 144, 56, 56]	288
ReLU6-30	[20, 144, 56, 56]	0
Conv2d-31	[20, 144, 28, 28]	1,296
BatchNorm2d-32	[20, 144, 28, 28]	288
ReLU6-33	[20, 144, 28, 28]	0
Conv2d-34	[20, 32, 28, 28]	4,608
BatchNorm2d-35	[20, 32, 28, 28]	64
InvertedResidual-36	[20, 32, 28, 28]	0
Conv2d-37	[20, 192, 28, 28]	6,144
BatchNorm2d-38	[20, 192, 28, 28]	384
ReLU6-39	[20, 192, 28, 28]	0
Conv2d-40	[20, 192, 28, 28]	1,728
BatchNorm2d-41	[20, 192, 28, 28]	384
ReLU6-42	[20, 192, 28, 28]	0
Conv2d-43	[20, 32, 28, 28]	6,144
BatchNorm2d-44	[20, 32, 28, 28]	64
InvertedResidual-45	[20, 32, 28, 28]	0
Conv2d-46	[20, 192, 28, 28]	6,144
BatchNorm2d-47	[20, 192, 28, 28]	384
ReLU6-48	[20, 192, 28, 28]	0
Conv2d-49	[20, 192, 28, 28]	1,728
BatchNorm2d-50	[20, 192, 28, 28]	384
ReLU6-51	[20, 192, 28, 28]	0
Conv2d-52	[20, 32, 28, 28]	6,144
BatchNorm2d-53	[20, 32, 28, 28]	64
InvertedResidual-54	[20, 32, 28, 28]	0
Conv2d-55	[20, 192, 28, 28]	6,144
BatchNorm2d-56	[20, 192, 28, 28]	384
ReLU6-57	[20, 192, 28, 28]	0
Conv2d-58	[20, 192, 14, 14]	1,728

BatchNorm2d-59	[20, 192, 14, 14]	384
ReLU6-60	[20, 192, 14, 14]	0
Conv2d-61	[20, 64, 14, 14]	12,288
BatchNorm2d-62	[20, 64, 14, 14]	128
InvertedResidual-63	[20, 64, 14, 14]	0
Conv2d-64	[20, 384, 14, 14]	24,576
BatchNorm2d-65	[20, 384, 14, 14]	768
ReLU6-66	[20, 384, 14, 14]	0
Conv2d-67	[20, 384, 14, 14]	3,456
BatchNorm2d-68	[20, 384, 14, 14]	768
ReLU6-69	[20, 384, 14, 14]	0
Conv2d-70	[20, 64, 14, 14]	24,576
BatchNorm2d-71	[20, 64, 14, 14]	128
InvertedResidual-72	[20, 64, 14, 14]	0
Conv2d-73	[20, 384, 14, 14]	24,576
BatchNorm2d-74	[20, 384, 14, 14]	768
ReLU6-75	[20, 384, 14, 14]	0
Conv2d-76	[20, 384, 14, 14]	3,456
BatchNorm2d-77	[20, 384, 14, 14]	768
ReLU6-78	[20, 384, 14, 14]	0
Conv2d-79	[20, 64, 14, 14]	24,576
BatchNorm2d-80	[20, 64, 14, 14]	128
InvertedResidual-81	[20, 64, 14, 14]	0
Conv2d-82	[20, 384, 14, 14]	24,576
BatchNorm2d-83	[20, 384, 14, 14]	768
ReLU6-84	[20, 384, 14, 14]	0
Conv2d-85	[20, 384, 14, 14]	3,456
BatchNorm2d-86	[20, 384, 14, 14]	768
ReLU6-87	[20, 384, 14, 14]	0
Conv2d-88	[20, 64, 14, 14]	24,576
BatchNorm2d-89	[20, 64, 14, 14]	128
InvertedResidual-90	[20, 64, 14, 14]	0
Conv2d-91	[20, 384, 14, 14]	24,576
BatchNorm2d-92	[20, 384, 14, 14]	768
ReLU6-93	[20, 384, 14, 14]	0
Conv2d-94	[20, 384, 14, 14]	3,456
BatchNorm2d-95	[20, 384, 14, 14]	768
ReLU6-96	[20, 384, 14, 14]	0
Conv2d-97	[20, 96, 14, 14]	36,864
BatchNorm2d-98	[20, 96, 14, 14]	192
InvertedResidual-99	[20, 96, 14, 14]	0
Conv2d-100	[20, 576, 14, 14]	55,296
BatchNorm2d-101	[20, 576, 14, 14]	1,152
ReLU6-102	[20, 576, 14, 14]	0
Conv2d-103	[20, 576, 14, 14]	5,184
BatchNorm2d-104	[20, 576, 14, 14]	1,152
ReLU6-105	[20, 576, 14, 14]	0
Conv2d-106	[20, 96, 14, 14]	55,296
BatchNorm2d-107	[20, 96, 14, 14]	192
InvertedResidual-108	[20, 96, 14, 14]	0
Conv2d-109	[20, 576, 14, 14]	55,296
BatchNorm2d-110	[20, 576, 14, 14]	1,152
ReLU6-111	[20, 576, 14, 14]	0
Conv2d-112	[20, 576, 14, 14]	5,184
BatchNorm2d-113	[20, 576, 14, 14]	1,152
ReLU6-114	[20, 576, 14, 14]	0
Conv2d-115	[20, 96, 14, 14]	55,296
BatchNorm2d-116	[20, 96, 14, 14]	192
InvertedResidual-117	[20, 96, 14, 14]	0
Conv2d-118	[20, 576, 14, 14]	55,296
BatchNorm2d-119	[20, 576, 14, 14]	1,152

ReLU6-120	[20, 576, 14, 14]	0
Conv2d-121	[20, 576, 7, 7]	5,184
BatchNorm2d-122	[20, 576, 7, 7]	1,152
ReLU6-123	[20, 576, 7, 7]	0
Conv2d-124	[20, 160, 7, 7]	92,160
BatchNorm2d-125	[20, 160, 7, 7]	320
InvertedResidual-126	[20, 160, 7, 7]	0
Conv2d-127	[20, 960, 7, 7]	153,600
BatchNorm2d-128	[20, 960, 7, 7]	1,920
ReLU6-129	[20, 960, 7, 7]	0
Conv2d-130	[20, 960, 7, 7]	8,640
BatchNorm2d-131	[20, 960, 7, 7]	1,920
ReLU6-132	[20, 960, 7, 7]	0
Conv2d-133	[20, 160, 7, 7]	153,600
BatchNorm2d-134	[20, 160, 7, 7]	320
InvertedResidual-135	[20, 160, 7, 7]	0
Conv2d-136	[20, 960, 7, 7]	153,600
BatchNorm2d-137	[20, 960, 7, 7]	1,920
ReLU6-138	[20, 960, 7, 7]	0
Conv2d-139	[20, 960, 7, 7]	8,640
BatchNorm2d-140	[20, 960, 7, 7]	1,920
ReLU6-141	[20, 960, 7, 7]	0
Conv2d-142	[20, 160, 7, 7]	153,600
BatchNorm2d-143	[20, 160, 7, 7]	320
InvertedResidual-144	[20, 160, 7, 7]	0
Conv2d-145	[20, 960, 7, 7]	153,600
BatchNorm2d-146	[20, 960, 7, 7]	1,920
ReLU6-147	[20, 960, 7, 7]	0
Conv2d-148	[20, 960, 7, 7]	8,640
BatchNorm2d-149	[20, 960, 7, 7]	1,920
ReLU6-150	[20, 960, 7, 7]	0
Conv2d-151	[20, 320, 7, 7]	307,200
BatchNorm2d-152	[20, 320, 7, 7]	640
InvertedResidual-153	[20, 320, 7, 7]	0
Conv2d-154	[20, 1280, 7, 7]	409,600
BatchNorm2d-155	[20, 1280, 7, 7]	2,560
ReLU6-156	[20, 1280, 7, 7]	0
Dropout-157	[20, 1280]	0
Linear-158	[20, 12]	15,372

=====
Total params: 2,239,244

Trainable params: 15,372

Non-trainable params: 2,223,872

Input size (MB): 11.48

Forward/backward pass size (MB): 3057.19

Params size (MB): 8.54

Estimated Total Size (MB): 3077.22

In [0]:

```
# criterion and optimzier
from torch import optim
criterion_mobilev2h = nn.CrossEntropyLoss()
optimizer_mobilev2h = optim.Adam(model_transfer_mobilev2.classifier.parameters
(), lr=0.001)
```

In [0]:

```
import torch
import shutil
def save_ckp(state, is_best, checkpoint_path, best_model_path):
    f_path = checkpoint_path
    torch.save(state, f_path)
    if is_best:
        best_fpath = best_model_path
        shutil.copyfile(f_path, best_fpath)
```

In [0]:

```
def load_ckp(checkpoint_fpath, model, optimizer):
    """
    checkpoint_path: path to save checkpoint
    model: model that we want to load checkpoint parameters into
    optimizer: optimizer we defined in previous training
    """
    # load check point
    checkpoint = torch.load(checkpoint_fpath)
    # initialize state_dict from checkpoint to model
    model.load_state_dict(checkpoint['state_dict'])
    # initialize optimizer from checkpoint to optimizer
    optimizer.load_state_dict(checkpoint['optimizer'])
    # initialize valid_loss_min from checkpoint to valid_loss_min
    valid_loss_min = checkpoint['valid_loss_min']
    # initialize train_loss_coll from checkpoint to train_loss_coll
    train_loss_coll = checkpoint['train_loss_coll']
    # initialize valid_loss_coll from checkpoint to valid_loss_coll
    valid_loss_coll = checkpoint['valid_loss_coll']
    # return model, optimizer, epoch value, min validation loss
    return model, optimizer, checkpoint['epoch'], valid_loss_min, train_loss_coll, valid_loss_coll
```


In [0]:

```

def train(start_epochs, n_epochs, valid_loss_min_input, loaders, model, optimizer, criterion, use_cuda, checkpoint_path, best_model_path):
    """returns trained model"""
    # initialize tracker for minimum validation loss
    valid_loss_min = valid_loss_min_input

    train_loss_coll = []
    valid_loss_coll = []

    for epoch in range(start_epochs, n_epochs+1):
        # initialize variables to monitor training and validation loss
        train_loss = 0.0
        train_correct = 0.
        train_total = 0.

        valid_loss = 0.0
        valid_correct = 0.
        valid_total = 0.
        #####
        # train the model #
        #####
        model.train()
        for batch_idx, (train_data, train_target) in enumerate(loaders['train']
    ):
        # move tensors to GPU if CUDA is available
        if use_cuda:
            train_data, train_target = train_data.cuda(), train_target.cuda
        ()
        # clear the gradients of all optimized variables
        optimizer.zero_grad()
        # forward pass: compute predicted outputs by passing inputs to the model
        train_output = model(train_data)
        # calculate the batch loss
        loss = criterion(train_output, train_target)
        # backward pass: compute gradient of the loss with respect to model parameters
        loss.backward()
        # perform a single optimization step (parameter update)
        optimizer.step()
        # update total training loss
        train_loss += loss.item()*train_data.size(0)
        # convert output logits to predicted class
        train_pred = train_output.data.max(1, keepdim=True)[1]
        # compare predictions to true label
        train_correct += np.sum(np.squeeze(train_pred.eq(train_target.data.view_as(train_pred))).cpu().numpy())
        train_total += train_data.size(0)

        ## find the loss and update the model parameters accordingly
        ## record the average training loss, using something like
        ## train_loss = train_loss + ((1 / (batch_idx + 1)) * (loss.data - t
rain_loss))

        #####
        # validate the model #
        #####
        model.eval()
        for val_batch_idx, (val_data, val_target) in enumerate(loaders['valid']

```

```

}):
    # move to GPU
    if use_cuda:
        val_data, val_target = val_data.cuda(), val_target.cuda()
    # forward pass: compute predicted outputs by passing inputs to the model
    val_output = model(val_data)
    # calculate the batch loss
    val_loss = criterion(val_output, val_target)
    # update total validation loss
    valid_loss += val_loss.item()*val_data.size(0)
    # convert output logits to predicted class
    valid_pred = val_output.data.max(1, keepdim=True)[1]
    # compare predictions to true label
    valid_correct += np.sum(np.squeeze(valid_pred.eq(val_target.data.view_as(valid_pred))).cpu().numpy())
    valid_total += val_data.size(0)

    # calculate average losses
    train_loss = train_loss/len(loaders['train'].dataset)
    valid_loss = valid_loss/len(loaders['valid'].dataset)

    train_acc = 100. * train_correct / train_total
    valid_acc = 100. * valid_correct / valid_total

    # print training/validation statistics
    print('Epoch: {} \tTrain Loss: {:.6f} \tValid Loss: {:.6f} \tTrain Acc: {:.2f} \tValid Acc: {:.2f}'.format(
        epoch,
        train_loss,
        valid_loss,
        train_acc,
        valid_acc))

    train_loss_coll.append(train_loss)
    valid_loss_coll.append(valid_loss)

    checkpoint = {
        'epoch': epoch + 1,
        'valid_loss_min': valid_loss,
        'state_dict': model.state_dict(),
        'optimizer': optimizer.state_dict(),
        'train_loss_coll': train_loss_coll,
        'valid_loss_coll': valid_loss_coll,
    }

    save_ckp(checkpoint, False, checkpoint_path, best_model_path)

    ## Save the model if validation loss has decreased
    if valid_loss <= valid_loss_min:
        print('Validation loss decreased ({:.6f} --> {:.6f}). Saving model ...'.format(
            valid_loss_min, valid_loss
        ))
        save_ckp(checkpoint, True, checkpoint_path, best_model_path)
        valid_loss_min = valid_loss

    # return trained model
    return model, train_loss_coll, valid_loss_coll

```

In [0]:

```
model_transfer_mobilev2, train_loss_coll_mobilev2, valid_loss_coll_mobilev2 = train(1, 25, np.Inf, data_loaders, model_transfer_mobilev2, optimizer_mobilev2h, criterion_mobilev2h, use_cuda, "./checkpoint/current_checkpoint_mobilev2.pt", "./best_model/best_model_mobilev2.pt")
```

```
/usr/local/lib/python3.6/dist-packages/PIL/Image.py:989: UserWarning: Palette images with Transparency expressed in bytes should be converted to RGBA images
```

```
"Palette images with Transparency expressed in bytes should be "
```

```

Epoch: 1      Train Loss: 1.568911    Valid Loss: 0.584078    Trai
n Acc: 60.85    Valid Acc: 81.14
Validation loss decreased (inf --> 0.584078). Saving model ...
Epoch: 2      Train Loss: 0.536801    Valid Loss: 0.375548    Trai
n Acc: 83.06    Valid Acc: 88.87
Validation loss decreased (0.584078 --> 0.375548). Saving model ...
Epoch: 3      Train Loss: 0.425447    Valid Loss: 0.333101    Trai
n Acc: 86.02    Valid Acc: 87.94
Validation loss decreased (0.375548 --> 0.333101). Saving model ...
Epoch: 4      Train Loss: 0.395263    Valid Loss: 0.294529    Trai
n Acc: 86.79    Valid Acc: 89.03
Validation loss decreased (0.333101 --> 0.294529). Saving model ...
Epoch: 5      Train Loss: 0.367737    Valid Loss: 0.295641    Trai
n Acc: 87.66    Valid Acc: 89.49
Epoch: 6      Train Loss: 0.350772    Valid Loss: 0.270243    Trai
n Acc: 88.94    Valid Acc: 91.65
Validation loss decreased (0.294529 --> 0.270243). Saving model ...
Epoch: 7      Train Loss: 0.334747    Valid Loss: 0.251904    Trai
n Acc: 88.98    Valid Acc: 91.81
Validation loss decreased (0.270243 --> 0.251904). Saving model ...
Epoch: 8      Train Loss: 0.331458    Valid Loss: 0.287743    Trai
n Acc: 88.80    Valid Acc: 91.19
Epoch: 9      Train Loss: 0.316900    Valid Loss: 0.267718    Trai
n Acc: 89.60    Valid Acc: 89.64
Epoch: 10     Train Loss: 0.310306    Valid Loss: 0.252412    Trai
n Acc: 89.53    Valid Acc: 92.27
Epoch: 11     Train Loss: 0.311519    Valid Loss: 0.258386    Trai
n Acc: 89.39    Valid Acc: 91.81
Epoch: 12     Train Loss: 0.293428    Valid Loss: 0.277846    Trai
n Acc: 90.29    Valid Acc: 89.80
Epoch: 13     Train Loss: 0.307031    Valid Loss: 0.261827    Trai
n Acc: 89.86    Valid Acc: 90.73
Epoch: 14     Train Loss: 0.289199    Valid Loss: 0.257734    Trai
n Acc: 89.86    Valid Acc: 91.50
Epoch: 15     Train Loss: 0.284032    Valid Loss: 0.259016    Trai
n Acc: 90.78    Valid Acc: 91.04
Epoch: 16     Train Loss: 0.310446    Valid Loss: 0.246643    Trai
n Acc: 89.62    Valid Acc: 91.34
Validation loss decreased (0.251904 --> 0.246643). Saving model ...
Epoch: 17     Train Loss: 0.287328    Valid Loss: 0.253140    Trai
n Acc: 90.67    Valid Acc: 91.34
Epoch: 18     Train Loss: 0.306613    Valid Loss: 0.265003    Trai
n Acc: 89.90    Valid Acc: 90.88
Epoch: 19     Train Loss: 0.300017    Valid Loss: 0.243602    Trai
n Acc: 89.81    Valid Acc: 91.65
Validation loss decreased (0.246643 --> 0.243602). Saving model ...
Epoch: 20     Train Loss: 0.292613    Valid Loss: 0.283308    Trai
n Acc: 90.22    Valid Acc: 89.80
Epoch: 21     Train Loss: 0.282531    Valid Loss: 0.331514    Trai
n Acc: 90.71    Valid Acc: 88.10
Epoch: 22     Train Loss: 0.295721    Valid Loss: 0.280837    Trai
n Acc: 89.94    Valid Acc: 90.11
Epoch: 23     Train Loss: 0.278208    Valid Loss: 0.267521    Trai
n Acc: 91.07    Valid Acc: 91.96
Epoch: 24     Train Loss: 0.275581    Valid Loss: 0.250385    Trai
n Acc: 91.00    Valid Acc: 91.50
Epoch: 25     Train Loss: 0.300611    Valid Loss: 0.273916    Trai
n Acc: 90.22    Valid Acc: 90.42

```

In [4]:

```
# these arrays created manually based on training above because the value not saved once google colab loaded again or interrupted
# I make change to save and load method to save train_loss_coll and valid_loss_coll so that
# later on we don't need to create this manually anymore
train_loss_coll_mobilev2_loaded = []
train_loss_coll_mobilev2_loaded.append(1.568911)
train_loss_coll_mobilev2_loaded.append(0.536801)
train_loss_coll_mobilev2_loaded.append(0.425447)
train_loss_coll_mobilev2_loaded.append(0.395263)
train_loss_coll_mobilev2_loaded.append(0.367737)
train_loss_coll_mobilev2_loaded.append(0.350772)
train_loss_coll_mobilev2_loaded.append(0.334747)
train_loss_coll_mobilev2_loaded.append(0.331458)
train_loss_coll_mobilev2_loaded.append(0.316900)
train_loss_coll_mobilev2_loaded.append(0.310306)
train_loss_coll_mobilev2_loaded.append(0.311519)
train_loss_coll_mobilev2_loaded.append(0.293428)
train_loss_coll_mobilev2_loaded.append(0.307031)
train_loss_coll_mobilev2_loaded.append(0.289199)
train_loss_coll_mobilev2_loaded.append(0.284032)
train_loss_coll_mobilev2_loaded.append(0.310446)
train_loss_coll_mobilev2_loaded.append(0.287328)
train_loss_coll_mobilev2_loaded.append(0.306613)
train_loss_coll_mobilev2_loaded.append(0.300017)
train_loss_coll_mobilev2_loaded.append(0.292613)
train_loss_coll_mobilev2_loaded.append(0.282531)
train_loss_coll_mobilev2_loaded.append(0.295721)
train_loss_coll_mobilev2_loaded.append(0.278208)
train_loss_coll_mobilev2_loaded.append(0.275581)
train_loss_coll_mobilev2_loaded.append(0.300611)

valid_loss_coll_mobilev2_loaded = []
valid_loss_coll_mobilev2_loaded.append(0.584078)
valid_loss_coll_mobilev2_loaded.append(0.375548)
valid_loss_coll_mobilev2_loaded.append(0.333101)
valid_loss_coll_mobilev2_loaded.append(0.294529)
valid_loss_coll_mobilev2_loaded.append(0.295641)
valid_loss_coll_mobilev2_loaded.append(0.270243)
valid_loss_coll_mobilev2_loaded.append(0.251904)
valid_loss_coll_mobilev2_loaded.append(0.287743)
valid_loss_coll_mobilev2_loaded.append(0.267718)
valid_loss_coll_mobilev2_loaded.append(0.252412)
valid_loss_coll_mobilev2_loaded.append(0.258386)
valid_loss_coll_mobilev2_loaded.append(0.277846)
valid_loss_coll_mobilev2_loaded.append(0.261827)
valid_loss_coll_mobilev2_loaded.append(0.257734)
valid_loss_coll_mobilev2_loaded.append(0.259016)
valid_loss_coll_mobilev2_loaded.append(0.246643)
valid_loss_coll_mobilev2_loaded.append(0.253140)
valid_loss_coll_mobilev2_loaded.append(0.265003)
valid_loss_coll_mobilev2_loaded.append(0.243602)
valid_loss_coll_mobilev2_loaded.append(0.283308)
valid_loss_coll_mobilev2_loaded.append(0.331514)
valid_loss_coll_mobilev2_loaded.append(0.280837)
valid_loss_coll_mobilev2_loaded.append(0.267521)
valid_loss_coll_mobilev2_loaded.append(0.250385)
valid_loss_coll_mobilev2_loaded.append(0.273916)
```

```
print("Train Loss: {:.6f}".format(valid_loss_coll_mobilev2_loaded[1]))
```

Train Loss: 0.375548

In [103]:

```
print(model_transfer_mobilev2)
```



```

MobileNetV2(
  (features): Sequential(
    (0): ConvBNReLU(
      (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(2, 2), padding=
(1, 1), bias=False)
      (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, tra
ck_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (1): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), pad
ding=(1, 1), groups=32, bias=False)
          (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): Conv2d(32, 16, kernel_size=(1, 1), stride=(1, 1), bias=
False)
        (2): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
      )
    )
    (2): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(16, 96, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(96, 96, kernel_size=(3, 3), stride=(2, 2), pad
ding=(1, 1), groups=96, bias=False)
          (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(96, 24, kernel_size=(1, 1), stride=(1, 1), bias=
False)
        (3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
      )
    )
    (3): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(24, 144, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(144, 144, kernel_size=(3, 3), stride=(1, 1), p
adding=(1, 1), groups=144, bias=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)

```

```

    )
    (2): Conv2d(144, 24, kernel_size=(1, 1), stride=(1, 1), bias
=False)
    (3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
    )
    )
    (4): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(24, 144, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(144, 144, kernel_size=(3, 3), stride=(2, 2), p
adding=(1, 1), groups=144, bias=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(144, 32, kernel_size=(1, 1), stride=(1, 1), bias
=False)
        (3): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
      )
    )
    (5): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(192, 192, kernel_size=(3, 3), stride=(1, 1), p
adding=(1, 1), groups=192, bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(192, 32, kernel_size=(1, 1), stride=(1, 1), bias
=False)
        (3): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
      )
    )
    (6): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(

```

```

        (0): Conv2d(192, 192, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=192, bias=False)
        (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU6(inplace=True)
    )
    (2): Conv2d(192, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    )
    (7): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(192, 192, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=192, bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(192, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (8): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=384, bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    )
    (9): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)

```

```

        (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU6(inplace=True)
    )
    (1): ConvBNReLU(
        (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=384, bias=False)
        (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU6(inplace=True)
    )
    (2): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    )
    (10): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=384, bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (11): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=384, bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(384, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
  )

```

```

(12): InvertedResidual(
  (conv): Sequential(
    (0): ConvBNReLU(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (1): ConvBNReLU(
      (0): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=576, bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (2): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(13): InvertedResidual(
  (conv): Sequential(
    (0): ConvBNReLU(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (1): ConvBNReLU(
      (0): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=576, bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (2): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(14): InvertedResidual(
  (conv): Sequential(
    (0): ConvBNReLU(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (1): ConvBNReLU(
      (0): Conv2d(576, 576, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=576, bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (2): Conv2d(576, 160, kernel_size=(1, 1), stride=(1, 1), bias=False)
  )
)

```

```

s=False)
    (3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    )
    )
    (15): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), b
ias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), p
adding=(1, 1), groups=960, bias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
        (3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
    )
    (16): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), b
ias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), p
adding=(1, 1), groups=960, bias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
        (3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
    )
    (17): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), b
ias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), p
adding=(1, 1), groups=960, bias=False)

```

```
        (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU6(inplace=True)
    )
    (2): Conv2d(960, 320, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): BatchNorm2d(320, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    )
    (18): ConvBNReLU(
      (0): Conv2d(320, 1280, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(1280, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
  )
  (classifier): Sequential(
    (0): Dropout(p=0.2, inplace=False)
    (1): Linear(in_features=1280, out_features=12, bias=True)
  )
)
```

In [105]:

```
model_transfer_mobilev2, train_loss_coll_mobilev2, valid_loss_coll_mobilev2 = train(1, 25, np.Inf, data_loaders, model_transfer_mobilev2, optimizer_mobilev2h, criterion_mobilev2h, use_cuda, "./checkpoint/current_checkpoint_mobilev2.pt", "./best_model/best_model_mobilev2.pt")
```



```
/usr/local/lib/python3.6/dist-packages/PIL/Image.py:989: UserWarning: Palette images with Transparency expressed in bytes should be converted to RGBA images
```

```
"Palette images with Transparency expressed in bytes should be "
```

```
Epoch: 1      Train Loss: 0.425292    Valid Loss: 0.295952    Train Acc: 86.61    Valid Acc: 89.34
Validation loss decreased (inf --> 0.295952). Saving model ...
Epoch: 2      Train Loss: 0.371286    Valid Loss: 0.274422    Train Acc: 87.65    Valid Acc: 91.19
Validation loss decreased (0.295952 --> 0.274422). Saving model ...
Epoch: 3      Train Loss: 0.352919    Valid Loss: 0.246933    Train Acc: 88.11    Valid Acc: 91.96
Validation loss decreased (0.274422 --> 0.246933). Saving model ...
Epoch: 4      Train Loss: 0.342864    Valid Loss: 0.231448    Train Acc: 89.00    Valid Acc: 92.74
Validation loss decreased (0.246933 --> 0.231448). Saving model ...
Epoch: 5      Train Loss: 0.321413    Valid Loss: 0.272789    Train Acc: 89.37    Valid Acc: 90.73
Epoch: 6      Train Loss: 0.299698    Valid Loss: 0.244372    Train Acc: 90.28    Valid Acc: 91.04
Epoch: 7      Train Loss: 0.314361    Valid Loss: 0.263974    Train Acc: 89.19    Valid Acc: 90.88
Epoch: 8      Train Loss: 0.302646    Valid Loss: 0.240826    Train Acc: 89.76    Valid Acc: 91.96
Epoch: 9      Train Loss: 0.295549    Valid Loss: 0.269712    Train Acc: 89.90    Valid Acc: 91.50
Epoch: 10     Train Loss: 0.294684    Valid Loss: 0.265772    Train Acc: 89.86    Valid Acc: 91.65
Epoch: 11     Train Loss: 0.290346    Valid Loss: 0.261724    Train Acc: 90.15    Valid Acc: 91.34
Epoch: 12     Train Loss: 0.297809    Valid Loss: 0.247449    Train Acc: 90.14    Valid Acc: 91.96
Epoch: 13     Train Loss: 0.298455    Valid Loss: 0.260202    Train Acc: 89.75    Valid Acc: 90.73
Epoch: 14     Train Loss: 0.303600    Valid Loss: 0.245402    Train Acc: 89.78    Valid Acc: 91.81
Epoch: 15     Train Loss: 0.299806    Valid Loss: 0.254209    Train Acc: 89.83    Valid Acc: 90.11
Epoch: 16     Train Loss: 0.293767    Valid Loss: 0.257622    Train Acc: 89.85    Valid Acc: 90.88
Epoch: 17     Train Loss: 0.296539    Valid Loss: 0.250410    Train Acc: 90.29    Valid Acc: 91.19
Epoch: 18     Train Loss: 0.288315    Valid Loss: 0.241241    Train Acc: 90.40    Valid Acc: 91.50
Epoch: 19     Train Loss: 0.281917    Valid Loss: 0.258978    Train Acc: 90.46    Valid Acc: 91.65
Epoch: 20     Train Loss: 0.293272    Valid Loss: 0.254144    Train Acc: 89.93    Valid Acc: 92.12
Epoch: 21     Train Loss: 0.286327    Valid Loss: 0.241950    Train Acc: 90.46    Valid Acc: 92.27
Epoch: 22     Train Loss: 0.297380    Valid Loss: 0.262114    Train Acc: 89.96    Valid Acc: 91.81
Epoch: 23     Train Loss: 0.275305    Valid Loss: 0.280302    Train Acc: 90.61    Valid Acc: 90.11
Epoch: 24     Train Loss: 0.282952    Valid Loss: 0.254441    Train Acc: 90.45    Valid Acc: 91.96
Epoch: 25     Train Loss: 0.273756    Valid Loss: 0.254767    Train Acc: 90.65    Valid Acc: 90.42
```

In [0]:

```
def test(loaders, model, criterion, use_cuda):

    # monitor test loss and accuracy
    test_loss = 0.
    correct = 0.
    total = 0.

    model.eval()
    for batch_idx, (data, target) in enumerate(loaders['test']):
        # move to GPU
        if use_cuda:
            data, target = data.cuda(), target.cuda()
        # forward pass: compute predicted outputs by passing inputs to the model
        output = model(data)
        # calculate the loss
        loss = criterion(output, target)
        # update average test loss
        test_loss = test_loss + ((1 / (batch_idx + 1)) * (loss.data - test_loss
    ))

        # convert output logits to predicted class
        pred = output.data.max(1, keepdim=True)[1]
        # compare predictions to true label
        correct += np.sum(np.squeeze(pred.eq(target.data.view_as(pred))).cpu().n
umpy())

        total += data.size(0)

    print('Test Loss: {:.6f}\n'.format(test_loss))

    print('\nTest Accuracy: %2d%% (%2d/%2d)' % (
        100. * correct / total, correct, total))
```

In [0]:

```
# load state of saved best model
ckp_path = './best_model/best_model_mobilev2.pt'
model_transfer_mobilev2, optimizer_mobilev2h, start_epoch, valid_loss_min, train
_loss_coll_mobilev2, valid_loss_coll_mobilev2 = load_ckpt(ckp_path, model_transfe
r_mobilev2, optimizer_mobilev2h)
```

In [115]:

```
print("model = ", model_transfer_mobilev2)
print("optimizer = ", optimizer_mobilev2h)
print("start_epoch = ", start_epoch)
print("valid_loss_min = ", valid_loss_min)
print("valid_loss_min = {:.6f}".format(valid_loss_min))
```

```

model = MobileNetV2(
    (features): Sequential(
      (0): ConvBNReLU(
        (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(2, 2), padding=
(1, 1), bias=False)
        (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, tra
ck_running_stats=True)
        (2): ReLU6(inplace=True)
      )
      (1): InvertedResidual(
        (conv): Sequential(
          (0): ConvBNReLU(
            (0): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), pad
ding=(1, 1), groups=32, bias=False)
            (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
          )
          (1): Conv2d(32, 16, kernel_size=(1, 1), stride=(1, 1), bias=
False)
          (2): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
        )
      )
      (2): InvertedResidual(
        (conv): Sequential(
          (0): ConvBNReLU(
            (0): Conv2d(16, 96, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
            (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
          )
          (1): ConvBNReLU(
            (0): Conv2d(96, 96, kernel_size=(3, 3), stride=(2, 2), pad
ding=(1, 1), groups=96, bias=False)
            (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            (2): ReLU6(inplace=True)
          )
          (2): Conv2d(96, 24, kernel_size=(1, 1), stride=(1, 1), bias=
False)
          (3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
        )
      )
      (3): InvertedResidual(
        (conv): Sequential(
          (0): ConvBNReLU(
            (0): Conv2d(24, 144, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
            (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
            (2): ReLU6(inplace=True)
          )
          (1): ConvBNReLU(
            (0): Conv2d(144, 144, kernel_size=(3, 3), stride=(1, 1), p
adding=(1, 1), groups=144, bias=False)
            (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
            (2): ReLU6(inplace=True)

```

```

    )
    (2): Conv2d(144, 24, kernel_size=(1, 1), stride=(1, 1), bias
=False)
    (3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
    )
    )
    (4): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(24, 144, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(144, 144, kernel_size=(3, 3), stride=(2, 2), p
adding=(1, 1), groups=144, bias=False)
          (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(144, 32, kernel_size=(1, 1), stride=(1, 1), bias
=False)
        (3): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
      )
    )
    (5): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(192, 192, kernel_size=(3, 3), stride=(1, 1), p
adding=(1, 1), groups=192, bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(192, 32, kernel_size=(1, 1), stride=(1, 1), bias
=False)
        (3): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
      )
    )
    (6): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(

```

```

        (0): Conv2d(192, 192, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=192, bias=False)
        (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU6(inplace=True)
    )
    (2): Conv2d(192, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    )
    (7): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(32, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(192, 192, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=192, bias=False)
          (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(192, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (8): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=384, bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    )
    (9): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)

```

```

        (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU6(inplace=True)
    )
    (1): ConvBNReLU(
        (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=384, bias=False)
        (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU6(inplace=True)
    )
    (2): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    )
    (10): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=384, bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(384, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (11): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(64, 384, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(384, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=384, bias=False)
          (1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(384, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
  )

```

```

(12): InvertedResidual(
  (conv): Sequential(
    (0): ConvBNReLU(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (1): ConvBNReLU(
      (0): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=576, bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (2): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(13): InvertedResidual(
  (conv): Sequential(
    (0): ConvBNReLU(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (1): ConvBNReLU(
      (0): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=576, bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (2): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(14): InvertedResidual(
  (conv): Sequential(
    (0): ConvBNReLU(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (1): ConvBNReLU(
      (0): Conv2d(576, 576, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=576, bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
    (2): Conv2d(576, 160, kernel_size=(1, 1), stride=(1, 1), bias=False)
  )
)

```



```

s=False)
    (3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    )
    )
    (15): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), b
ias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), p
adding=(1, 1), groups=960, bias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
        (3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
    )
    (16): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), b
ias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), p
adding=(1, 1), groups=960, bias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (2): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
        (3): BatchNorm2d(160, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
    )
    (17): InvertedResidual(
      (conv): Sequential(
        (0): ConvBNReLU(
          (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), b
ias=False)
          (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=Tru
e, track_running_stats=True)
          (2): ReLU6(inplace=True)
        )
        (1): ConvBNReLU(
          (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1), p
adding=(1, 1), groups=960, bias=False)

```

```

        (1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (2): ReLU6(inplace=True)
    )
    (2): Conv2d(960, 320, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (3): BatchNorm2d(320, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    )
    (18): ConvBNReLU(
      (0): Conv2d(320, 1280, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(1280, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6(inplace=True)
    )
  )
  (classifier): Sequential(
    (0): Dropout(p=0.2, inplace=False)
    (1): Linear(in_features=1280, out_features=12, bias=True)
  )
)
optimizer = Adam (
Parameter Group 0
  amsgrad: False
  betas: (0.9, 0.999)
  eps: 1e-08
  lr: 0.001
  weight_decay: 0
)
start_epoch = 5
valid_loss_min = 0.23144799944479738
valid_loss_min = 0.231448

```

Model ShuffleNetv2

In [0]:

```
model_shufflenetv2 = models.shufflenet_v2_x1_0(pretrained=True)
print(model_shufflenetv2)
```

```

ShuffleNetV2(
  (conv1): Sequential(
    (0): Conv2d(3, 24, kernel_size=(3, 3), stride=(2, 2), padding=
(1, 1), bias=False)
    (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track
_running_stats=True)
    (2): ReLU(inplace=True)
  )
  (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=
1, ceil_mode=False)
  (stage2): Sequential(
    (0): InvertedResidual(
      (branch1): Sequential(
        (0): Conv2d(24, 24, kernel_size=(3, 3), stride=(2, 2), paddi
ng=(1, 1), groups=24, bias=False)
        (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
        (2): Conv2d(24, 58, kernel_size=(1, 1), stride=(1, 1), bias=
False)
        (3): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
        (4): ReLU(inplace=True)
      )
      (branch2): Sequential(
        (0): Conv2d(24, 58, kernel_size=(1, 1), stride=(1, 1), bias=
False)
        (1): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
        (2): ReLU(inplace=True)
        (3): Conv2d(58, 58, kernel_size=(3, 3), stride=(2, 2), paddi
ng=(1, 1), groups=58, bias=False)
        (4): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
        (5): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1), bias=
False)
        (6): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
        (7): ReLU(inplace=True)
      )
    )
  )
  (1): InvertedResidual(
    (branch2): Sequential(
      (0): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1), bias=
False)
      (1): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(58, 58, kernel_size=(3, 3), stride=(1, 1), paddi
ng=(1, 1), groups=58, bias=False)
      (4): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
      (5): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1), bias=
False)
      (6): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
      (7): ReLU(inplace=True)
    )
  )
  (2): InvertedResidual(
    (branch2): Sequential(
      (0): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1), bias=

```

```

False)
    (1): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(58, 58, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=58, bias=False)
    (4): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
(3): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(58, 58, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=58, bias=False)
    (4): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
)
(stage3): Sequential(
  (0): InvertedResidual(
    (branch1): Sequential(
      (0): Conv2d(116, 116, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=116, bias=False)
      (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (3): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (4): ReLU(inplace=True)
    )
    (branch2): Sequential(
      (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=116, bias=False)
      (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)

```

```
(7): ReLU(inplace=True)
)
)
(1): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116, bias=False)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
(2): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116, bias=False)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
(3): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116, bias=False)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
(4): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
```

```

        (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (2): ReLU(inplace=True)
        (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116, bias=False)
        (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (7): ReLU(inplace=True)
    )
)
(5): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116, bias=False)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
(6): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116, bias=False)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
(7): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116, bias=False)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)

```

```

        (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (7): ReLU(inplace=True)
    )
)
(stage4): Sequential(
  (0): InvertedResidual(
    (branch1): Sequential(
      (0): Conv2d(232, 232, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=232, bias=False)
      (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (3): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (4): ReLU(inplace=True)
    )
    (branch2): Sequential(
      (0): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(232, 232, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=232, bias=False)
      (4): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (6): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (7): ReLU(inplace=True)
    )
  )
  (1): InvertedResidual(
    (branch2): Sequential(
      (0): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(232, 232, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=232, bias=False)
      (4): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (6): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (7): ReLU(inplace=True)
    )
  )
  (2): InvertedResidual(
    (branch2): Sequential(
      (0): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True,

```



```

track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(232, 232, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=232, bias=False)
    (4): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
(3): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(232, 232, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=232, bias=False)
    (4): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
)
(conv5): Sequential(
  (0): Conv2d(464, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (1): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (2): ReLU(inplace=True)
)
(fc): Linear(in_features=1024, out_features=1000, bias=True)
)

```

In [0]:

```

# Freeze parameters so we don't backprop through them
for param in model_shufflenetv2.parameters():
    param.requires_grad = False

n_inputs = model_shufflenetv2.fc.in_features
n_classes = len(class_names)
model_shufflenetv2.fc = nn.Linear(n_inputs, n_classes, bias = True)

# Unfreeze training for classifier layers
for param in model_shufflenetv2.fc.parameters():
    param.requires_grad = True

```

In [0]:

```
if use_cuda:  
    model_shufflenetv2 = model_shufflenetv2.cuda()
```

In [0]:

```
next(model_shufflenetv2.parameters()).is_cuda
```

Out[0]:

True

In [0]:

```
print(model_shufflenetv2)
```

```

ShuffleNetV2(
  (conv1): Sequential(
    (0): Conv2d(3, 24, kernel_size=(3, 3), stride=(2, 2), padding=
(1, 1), bias=False)
    (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track
_running_stats=True)
    (2): ReLU(inplace=True)
  )
  (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=
1, ceil_mode=False)
  (stage2): Sequential(
    (0): InvertedResidual(
      (branch1): Sequential(
        (0): Conv2d(24, 24, kernel_size=(3, 3), stride=(2, 2), paddi
ng=(1, 1), groups=24, bias=False)
        (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
        (2): Conv2d(24, 58, kernel_size=(1, 1), stride=(1, 1), bias=
False)
        (3): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
        (4): ReLU(inplace=True)
      )
      (branch2): Sequential(
        (0): Conv2d(24, 58, kernel_size=(1, 1), stride=(1, 1), bias=
False)
        (1): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
        (2): ReLU(inplace=True)
        (3): Conv2d(58, 58, kernel_size=(3, 3), stride=(2, 2), paddi
ng=(1, 1), groups=58, bias=False)
        (4): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
        (5): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1), bias=
False)
        (6): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
        (7): ReLU(inplace=True)
      )
    )
    (1): InvertedResidual(
      (branch2): Sequential(
        (0): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1), bias=
False)
        (1): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
        (2): ReLU(inplace=True)
        (3): Conv2d(58, 58, kernel_size=(3, 3), stride=(1, 1), paddi
ng=(1, 1), groups=58, bias=False)
        (4): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
        (5): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1), bias=
False)
        (6): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
        (7): ReLU(inplace=True)
      )
    )
    (2): InvertedResidual(
      (branch2): Sequential(
        (0): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1), bias=

```

```

False)
    (1): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(58, 58, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=58, bias=False)
    (4): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
(3): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(58, 58, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=58, bias=False)
    (4): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
)
(stage3): Sequential(
  (0): InvertedResidual(
    (branch1): Sequential(
      (0): Conv2d(116, 116, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=116, bias=False)
      (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (3): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (4): ReLU(inplace=True)
    )
    (branch2): Sequential(
      (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=116, bias=False)
      (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)

```

```
        (7): ReLU(inplace=True)
    )
)
(1): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116, bias=False)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
(2): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116, bias=False)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
(3): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116, bias=False)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
(4): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
```

```

        (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (2): ReLU(inplace=True)
        (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116, bias=False)
        (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (7): ReLU(inplace=True)
    )
)
(5): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116, bias=False)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
(6): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116, bias=False)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
(7): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116, bias=False)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)

```

```

        (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (7): ReLU(inplace=True)
    )
)
(stage4): Sequential(
  (0): InvertedResidual(
    (branch1): Sequential(
      (0): Conv2d(232, 232, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=232, bias=False)
      (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (3): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (4): ReLU(inplace=True)
    )
    (branch2): Sequential(
      (0): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(232, 232, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=232, bias=False)
      (4): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (6): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (7): ReLU(inplace=True)
    )
  )
  (1): InvertedResidual(
    (branch2): Sequential(
      (0): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(232, 232, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=232, bias=False)
      (4): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (6): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (7): ReLU(inplace=True)
    )
  )
  (2): InvertedResidual(
    (branch2): Sequential(
      (0): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True,

```



```

track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(232, 232, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=232, bias=False)
    (4): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
(3): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(232, 232, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=232, bias=False)
    (4): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
)
(conv5): Sequential(
  (0): Conv2d(464, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (1): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (2): ReLU(inplace=True)
)
(fc): Linear(in_features=1024, out_features=1000, bias=True)
)

```

In [0]:

```

# criterion and optimzier
from torch import optim
criterion_shufflenetv2 = nn.CrossEntropyLoss()
optimizer_shufflenetv2 = optim.Adam(model_shufflenetv2.fc.parameters(), lr=0.001)
)

```

In [0]:

```
model_shufflenetv2, train_loss_coll_shufflenetv2, valid_loss_coll_shufflenetv2 =  
train(1, 25, np.Inf, data_loaders, model_shufflenetv2, optimizer_shufflenetv2, c  
riterion_shufflenetv2, use_cuda, "./checkpoint/current_checkpoint_shufflenetv2.p  
t", "./best_model/best_model_shufflenetv2.pt")
```

```
/usr/local/lib/python3.6/dist-packages/PIL/Image.py:989: UserWarning: Palette images with Transparency expressed in bytes should be converted to RGBA images
```

```
"Palette images with Transparency expressed in bytes should be "
```

```
Epoch: 1      Train Loss: 7.136982    Valid Loss: 5.674315    Trai
n Acc: 2.02    Valid Acc: 3.55
Validation loss decreased (inf --> 5.674315). Saving model ...
Epoch: 2      Train Loss: 4.377129    Valid Loss: 4.338295    Trai
n Acc: 8.93    Valid Acc: 8.35
Validation loss decreased (5.674315 --> 4.338295). Saving model ...
Epoch: 3      Train Loss: 3.546407    Valid Loss: 3.608678    Trai
n Acc: 14.71   Valid Acc: 13.76
Validation loss decreased (4.338295 --> 3.608678). Saving model ...
Epoch: 4      Train Loss: 3.040731    Valid Loss: 3.122319    Trai
n Acc: 19.87   Valid Acc: 19.32
Validation loss decreased (3.608678 --> 3.122319). Saving model ...
Epoch: 5      Train Loss: 2.670282    Valid Loss: 2.714511    Trai
n Acc: 26.79   Valid Acc: 24.88
Validation loss decreased (3.122319 --> 2.714511). Saving model ...
Epoch: 6      Train Loss: 2.358133    Valid Loss: 2.405763    Trai
n Acc: 32.70   Valid Acc: 32.61
Validation loss decreased (2.714511 --> 2.405763). Saving model ...
Epoch: 7      Train Loss: 2.125495    Valid Loss: 2.157099    Trai
n Acc: 37.58   Valid Acc: 36.94
Validation loss decreased (2.405763 --> 2.157099). Saving model ...
Epoch: 8      Train Loss: 1.916587    Valid Loss: 1.982034    Trai
n Acc: 42.16   Valid Acc: 40.19
Validation loss decreased (2.157099 --> 1.982034). Saving model ...
Epoch: 9      Train Loss: 1.774717    Valid Loss: 1.813637    Trai
n Acc: 45.63   Valid Acc: 45.13
Validation loss decreased (1.982034 --> 1.813637). Saving model ...
Epoch: 10     Train Loss: 1.648507    Valid Loss: 1.685187    Trai
n Acc: 50.11   Valid Acc: 47.45
Validation loss decreased (1.813637 --> 1.685187). Saving model ...
Epoch: 11     Train Loss: 1.521270    Valid Loss: 1.577104    Trai
n Acc: 53.00   Valid Acc: 51.00
Validation loss decreased (1.685187 --> 1.577104). Saving model ...
Epoch: 12     Train Loss: 1.445714    Valid Loss: 1.497572    Trai
n Acc: 55.33   Valid Acc: 52.55
Validation loss decreased (1.577104 --> 1.497572). Saving model ...
Epoch: 13     Train Loss: 1.384934    Valid Loss: 1.404356    Trai
n Acc: 57.12   Valid Acc: 57.34
Validation loss decreased (1.497572 --> 1.404356). Saving model ...
Epoch: 14     Train Loss: 1.320485    Valid Loss: 1.302281    Trai
n Acc: 59.15   Valid Acc: 58.27
Validation loss decreased (1.404356 --> 1.302281). Saving model ...
Epoch: 15     Train Loss: 1.253882    Valid Loss: 1.266706    Trai
n Acc: 61.24   Valid Acc: 60.28
Validation loss decreased (1.302281 --> 1.266706). Saving model ...
Epoch: 16     Train Loss: 1.191871    Valid Loss: 1.218365    Trai
n Acc: 62.95   Valid Acc: 62.13
Validation loss decreased (1.266706 --> 1.218365). Saving model ...
Epoch: 17     Train Loss: 1.151210    Valid Loss: 1.164641    Trai
n Acc: 63.67   Valid Acc: 63.99
Validation loss decreased (1.218365 --> 1.164641). Saving model ...
Epoch: 18     Train Loss: 1.115675    Valid Loss: 1.093461    Trai
n Acc: 65.16   Valid Acc: 65.53
Validation loss decreased (1.164641 --> 1.093461). Saving model ...
Epoch: 19     Train Loss: 1.078923    Valid Loss: 1.061117    Trai
n Acc: 66.40   Valid Acc: 66.31
Validation loss decreased (1.093461 --> 1.061117). Saving model ...
Epoch: 20     Train Loss: 1.048755    Valid Loss: 1.036347    Trai
n Acc: 67.51   Valid Acc: 66.46
Validation loss decreased (1.061117 --> 1.036347). Saving model ...
Epoch: 21     Train Loss: 1.012537    Valid Loss: 0.994682    Trai
```

```
n Acc: 68.34    Valid Acc: 68.32
Validation loss decreased (1.036347 --> 0.994682).  Saving model ...
Epoch: 22      Train Loss: 0.991943    Valid Loss: 0.984856    Trai
n Acc: 68.93    Valid Acc: 68.01
Validation loss decreased (0.994682 --> 0.984856).  Saving model ...
Epoch: 23      Train Loss: 0.955231    Valid Loss: 0.936794    Trai
n Acc: 69.74    Valid Acc: 70.32
Validation loss decreased (0.984856 --> 0.936794).  Saving model ...
Epoch: 24      Train Loss: 0.946270    Valid Loss: 0.944247    Trai
n Acc: 70.81    Valid Acc: 69.40
Epoch: 25      Train Loss: 0.931014    Valid Loss: 0.908774    Trai
n Acc: 71.04    Valid Acc: 71.56
Validation loss decreased (0.936794 --> 0.908774).  Saving model ...
```

In [0]:

```
# these arrays created manually based on training above because the value not saved once google colab loaded again or interrupted
# I make change to save and load method to save train_loss_coll and valid_loss_coll so that
# later on we don't need to create this manually anymore
train_loss_coll_shufflenetv2_loaded = []
train_loss_coll_shufflenetv2_loaded.append(7.136982)
train_loss_coll_shufflenetv2_loaded.append(4.377129)
train_loss_coll_shufflenetv2_loaded.append(3.546407)
train_loss_coll_shufflenetv2_loaded.append(3.040731)
train_loss_coll_shufflenetv2_loaded.append(2.670282)
train_loss_coll_shufflenetv2_loaded.append(2.358133)
train_loss_coll_shufflenetv2_loaded.append(2.125495)
train_loss_coll_shufflenetv2_loaded.append(1.916587)
train_loss_coll_shufflenetv2_loaded.append(1.774717)
train_loss_coll_shufflenetv2_loaded.append(1.648507)
train_loss_coll_shufflenetv2_loaded.append(1.521270)
train_loss_coll_shufflenetv2_loaded.append(1.445714)
train_loss_coll_shufflenetv2_loaded.append(1.384934)
train_loss_coll_shufflenetv2_loaded.append(1.320485)
train_loss_coll_shufflenetv2_loaded.append(1.253882)
train_loss_coll_shufflenetv2_loaded.append(1.191871)
train_loss_coll_shufflenetv2_loaded.append(1.151210)
train_loss_coll_shufflenetv2_loaded.append(1.115675)
train_loss_coll_shufflenetv2_loaded.append(1.078923)
train_loss_coll_shufflenetv2_loaded.append(1.048755)
train_loss_coll_shufflenetv2_loaded.append(1.012537)
train_loss_coll_shufflenetv2_loaded.append(0.991943)
train_loss_coll_shufflenetv2_loaded.append(0.955231)
train_loss_coll_shufflenetv2_loaded.append(0.946270)
train_loss_coll_shufflenetv2_loaded.append(0.931014)

valid_loss_coll_shufflenetv2_loaded = []
valid_loss_coll_shufflenetv2_loaded.append(5.674315)
valid_loss_coll_shufflenetv2_loaded.append(4.338295)
valid_loss_coll_shufflenetv2_loaded.append(3.608678)
valid_loss_coll_shufflenetv2_loaded.append(3.122319)
valid_loss_coll_shufflenetv2_loaded.append(2.714511)
valid_loss_coll_shufflenetv2_loaded.append(2.405763)
valid_loss_coll_shufflenetv2_loaded.append(2.157099)
valid_loss_coll_shufflenetv2_loaded.append(1.982034)
valid_loss_coll_shufflenetv2_loaded.append(1.813637)
valid_loss_coll_shufflenetv2_loaded.append(1.685187)
valid_loss_coll_shufflenetv2_loaded.append(1.577104)
valid_loss_coll_shufflenetv2_loaded.append(1.497572)
valid_loss_coll_shufflenetv2_loaded.append(1.404356)
valid_loss_coll_shufflenetv2_loaded.append(1.302281)
valid_loss_coll_shufflenetv2_loaded.append(1.266706)
valid_loss_coll_shufflenetv2_loaded.append(1.218365)
valid_loss_coll_shufflenetv2_loaded.append(1.164641)
valid_loss_coll_shufflenetv2_loaded.append(1.093461)
valid_loss_coll_shufflenetv2_loaded.append(1.061117)
valid_loss_coll_shufflenetv2_loaded.append(1.036347)
valid_loss_coll_shufflenetv2_loaded.append(0.994682)
valid_loss_coll_shufflenetv2_loaded.append(0.984856)
valid_loss_coll_shufflenetv2_loaded.append(0.936794)
valid_loss_coll_shufflenetv2_loaded.append(0.944247)
valid_loss_coll_shufflenetv2_loaded.append(0.908774)
```

In [0]:

```
# load state of saved best model
ckp_path = './best_model/best_model_shufflenetv2.pt'
model_shufflenetv2, optimizer_shufflenetv2, start_epoch, valid_loss_min = load_ckp(ckp_path, model_shufflenetv2, optimizer_shufflenetv2)
```

In [0]:

```
print("model = ", model_shufflenetv2)
print("optimizer = ", optimizer_shufflenetv2)
print("start_epoch = ", start_epoch)
print("valid_loss_min = ", valid_loss_min)
print("valid_loss_min = {:.6f}".format(valid_loss_min))
```



```

model = ShuffleNetV2(
    (conv1): Sequential(
      (0): Conv2d(3, 24, kernel_size=(3, 3), stride=(2, 2), padding=
(1, 1), bias=False)
      (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track
_running_stats=True)
      (2): ReLU(inplace=True)
    )
    (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=
1, ceil_mode=False)
    (stage2): Sequential(
      (0): InvertedResidual(
        (branch1): Sequential(
          (0): Conv2d(24, 24, kernel_size=(3, 3), stride=(2, 2), paddi
ng=(1, 1), groups=24, bias=False)
          (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
          (2): Conv2d(24, 58, kernel_size=(1, 1), stride=(1, 1), bias=
False)
          (3): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
          (4): ReLU(inplace=True)
        )
        (branch2): Sequential(
          (0): Conv2d(24, 58, kernel_size=(1, 1), stride=(1, 1), bias=
False)
          (1): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
          (2): ReLU(inplace=True)
          (3): Conv2d(58, 58, kernel_size=(3, 3), stride=(2, 2), paddi
ng=(1, 1), groups=58, bias=False)
          (4): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
          (5): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1), bias=
False)
          (6): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
          (7): ReLU(inplace=True)
        )
      )
      (1): InvertedResidual(
        (branch2): Sequential(
          (0): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1), bias=
False)
          (1): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
          (2): ReLU(inplace=True)
          (3): Conv2d(58, 58, kernel_size=(3, 3), stride=(1, 1), paddi
ng=(1, 1), groups=58, bias=False)
          (4): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
          (5): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1), bias=
False)
          (6): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, t
rack_running_stats=True)
          (7): ReLU(inplace=True)
        )
      )
      (2): InvertedResidual(
        (branch2): Sequential(
          (0): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1), bias=

```

```

False)
    (1): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(58, 58, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=58, bias=False)
    (4): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
(3): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(58, 58, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=58, bias=False)
    (4): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(58, 58, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(58, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
)
(stage3): Sequential(
  (0): InvertedResidual(
    (branch1): Sequential(
      (0): Conv2d(116, 116, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=116, bias=False)
      (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (3): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (4): ReLU(inplace=True)
    )
    (branch2): Sequential(
      (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=116, bias=False)
      (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)

```

```
        (7): ReLU(inplace=True)
    )
)
(1): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116, bias=False)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
(2): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116, bias=False)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
(3): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116, bias=False)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
(4): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
```

```

        (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (2): ReLU(inplace=True)
        (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116, bias=False)
        (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (7): ReLU(inplace=True)
    )
)
(5): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116, bias=False)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
(6): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116, bias=False)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
(7): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(116, 116, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=116, bias=False)
    (4): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)

```

```

        (5): Conv2d(116, 116, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (6): BatchNorm2d(116, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (7): ReLU(inplace=True)
    )
)
(stage4): Sequential(
  (0): InvertedResidual(
    (branch1): Sequential(
      (0): Conv2d(232, 232, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=232, bias=False)
      (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (3): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (4): ReLU(inplace=True)
    )
    (branch2): Sequential(
      (0): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(232, 232, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=232, bias=False)
      (4): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (6): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (7): ReLU(inplace=True)
    )
  )
  (1): InvertedResidual(
    (branch2): Sequential(
      (0): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(232, 232, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=232, bias=False)
      (4): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (5): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (6): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (7): ReLU(inplace=True)
    )
  )
  (2): InvertedResidual(
    (branch2): Sequential(
      (0): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True,

```

```

track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(232, 232, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=232, bias=False)
    (4): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
(3): InvertedResidual(
  (branch2): Sequential(
    (0): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(232, 232, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=232, bias=False)
    (4): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Conv2d(232, 232, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (6): BatchNorm2d(232, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
  )
)
)
(conv5): Sequential(
  (0): Conv2d(464, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (1): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (2): ReLU(inplace=True)
)
(fc): Linear(in_features=1024, out_features=1000, bias=True)
)
optimizer = Adam (
Parameter Group 0
  amsgrad: False
  betas: (0.9, 0.999)
  eps: 1e-08
  lr: 0.001
  weight_decay: 0
)
start_epoch = 26
valid_loss_min = 0.9087736422348612
valid_loss_min = 0.908774

```

Model MNASNET

In [0]:

```
mnasnet = models.mnasnet1_0(pretrained=True)
```

In [0]:

```
print(mnasnet)
```

```

MNASNet(
  (layers): Sequential(
    (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(2, 2), padding=
(1, 1), bias=False)
    (1): BatchNorm2d(32, eps=1e-05, momentum=0.000299999999999996696,
affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), groups=32, bias=False)
    (4): BatchNorm2d(32, eps=1e-05, momentum=0.000299999999999996696,
affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(32, 16, kernel_size=(1, 1), stride=(1, 1), bias=Fals
e)
    (7): BatchNorm2d(16, eps=1e-05, momentum=0.000299999999999996696,
affine=True, track_running_stats=True)
    (8): Sequential(
      (0): _InvertedResidual(
        (layers): Sequential(
          (0): Conv2d(16, 48, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (1): BatchNorm2d(48, eps=1e-05, momentum=0.000299999999999
96696, affine=True, track_running_stats=True)
          (2): ReLU(inplace=True)
          (3): Conv2d(48, 48, kernel_size=(3, 3), stride=(2, 2), pad
ding=(1, 1), groups=48, bias=False)
          (4): BatchNorm2d(48, eps=1e-05, momentum=0.000299999999999
96696, affine=True, track_running_stats=True)
          (5): ReLU(inplace=True)
          (6): Conv2d(48, 24, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (7): BatchNorm2d(24, eps=1e-05, momentum=0.000299999999999
96696, affine=True, track_running_stats=True)
        )
      )
      (1): _InvertedResidual(
        (layers): Sequential(
          (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (1): BatchNorm2d(72, eps=1e-05, momentum=0.000299999999999
96696, affine=True, track_running_stats=True)
          (2): ReLU(inplace=True)
          (3): Conv2d(72, 72, kernel_size=(3, 3), stride=(1, 1), pad
ding=(1, 1), groups=72, bias=False)
          (4): BatchNorm2d(72, eps=1e-05, momentum=0.000299999999999
96696, affine=True, track_running_stats=True)
          (5): ReLU(inplace=True)
          (6): Conv2d(72, 24, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (7): BatchNorm2d(24, eps=1e-05, momentum=0.000299999999999
96696, affine=True, track_running_stats=True)
        )
      )
      (2): _InvertedResidual(
        (layers): Sequential(
          (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (1): BatchNorm2d(72, eps=1e-05, momentum=0.000299999999999
96696, affine=True, track_running_stats=True)
          (2): ReLU(inplace=True)
          (3): Conv2d(72, 72, kernel_size=(3, 3), stride=(1, 1), pad

```



```

ding=(1, 1), groups=72, bias=False)
    (4): BatchNorm2d(72, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(72, 24, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
    (7): BatchNorm2d(24, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
  )
)
(9): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
      (1): BatchNorm2d(72, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(72, 72, kernel_size=(5, 5), stride=(2, 2), pad
ding=(2, 2), groups=72, bias=False)
      (4): BatchNorm2d(72, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(72, 40, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
      (7): BatchNorm2d(40, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(120, 120, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=120, bias=False)
      (4): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(120, 40, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(40, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (2): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(120, 120, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=120, bias=False)
      (4): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(120, 40, kernel_size=(1, 1), stride=(1, 1), bi

```

```

as=False)
    (7): BatchNorm2d(40, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
)
(10): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(40, 240, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(240, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(240, 240, kernel_size=(5, 5), stride=(2, 2), p
adding=(2, 2), groups=240, bias=False)
      (4): BatchNorm2d(240, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(240, 80, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(80, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(480, 480, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=480, bias=False)
      (4): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(480, 80, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(80, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (2): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(480, 480, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=480, bias=False)
      (4): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(480, 80, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(80, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
)

```

```

)
(11): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(480, 480, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=480, bias=False)
      (4): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(480, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (7): BatchNorm2d(96, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=576, bias=False)
      (4): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (7): BatchNorm2d(96, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
    )
  )
)
(12): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(576, 576, kernel_size=(5, 5), stride=(2, 2), padding=(2, 2), groups=576, bias=False)
      (4): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(576, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (7): BatchNorm2d(192, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),

```

```

bias=False)
    (1): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
    (4): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (7): BatchNorm2d(192, eps=1e-05, momentum=0.000299999999999
996696, affine=True, track_running_stats=True)
    )
    )
    (2): _InvertedResidual(
    (layers): Sequential(
    (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (1): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
    (4): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (7): BatchNorm2d(192, eps=1e-05, momentum=0.000299999999999
996696, affine=True, track_running_stats=True)
    )
    )
    (3): _InvertedResidual(
    (layers): Sequential(
    (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (1): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
    (4): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (7): BatchNorm2d(192, eps=1e-05, momentum=0.000299999999999
996696, affine=True, track_running_stats=True)
    )
    )
    )
    (13): Sequential(
    (0): _InvertedResidual(
    (layers): Sequential(
    (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (1): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(1152, 1152, kernel_size=(3, 3), stride=(1, 1),

```

```
padding=(1, 1), groups=1152, bias=False)
    (4): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(1152, 320, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (7): BatchNorm2d(320, eps=1e-05, momentum=0.000299999999999
996696, affine=True, track_running_stats=True)
    )
    )
    (14): Conv2d(320, 1280, kernel_size=(1, 1), stride=(1, 1), bias=
False)
    (15): BatchNorm2d(1280, eps=1e-05, momentum=0.000299999999999996
96, affine=True, track_running_stats=True)
    (16): ReLU(inplace=True)
    )
    (classifier): Sequential(
      (0): Dropout(p=0.2, inplace=True)
      (1): Linear(in_features=1280, out_features=1000, bias=True)
    )
  )
)
```

In [0]:

```
# Freeze parameters so we don't backprop through them
for param in mnasnet.parameters():
    param.requires_grad = False

n_inputs = mnasnet.classifier[1].in_features
n_classes = len(class_names)
mnasnet.classifier = nn.Sequential(
    nn.Dropout(0.2, inplace = True),
    nn.Linear(n_inputs, n_classes, bias = True))

# Unfreeze training for classifier layers
for param in mnasnet.classifier.parameters():
    param.requires_grad = True
```

In [0]:

```
if use_cuda:
    mnasnet = mnasnet.cuda()
```

In [133]:

```
next(mnasnet.parameters()).is_cuda
```

Out[133]:

True

In [134]:

```
print(mnasnet)
```

```

MNASNet(
  (layers): Sequential(
    (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(2, 2), padding=
(1, 1), bias=False)
    (1): BatchNorm2d(32, eps=1e-05, momentum=0.00029999999999999696,
affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), groups=32, bias=False)
    (4): BatchNorm2d(32, eps=1e-05, momentum=0.00029999999999999696,
affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(32, 16, kernel_size=(1, 1), stride=(1, 1), bias=Fals
e)
    (7): BatchNorm2d(16, eps=1e-05, momentum=0.00029999999999999696,
affine=True, track_running_stats=True)
    (8): Sequential(
      (0): _InvertedResidual(
        (layers): Sequential(
          (0): Conv2d(16, 48, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (1): BatchNorm2d(48, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          (2): ReLU(inplace=True)
          (3): Conv2d(48, 48, kernel_size=(3, 3), stride=(2, 2), pad
ding=(1, 1), groups=48, bias=False)
          (4): BatchNorm2d(48, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          (5): ReLU(inplace=True)
          (6): Conv2d(48, 24, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (7): BatchNorm2d(24, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
        )
      )
      (1): _InvertedResidual(
        (layers): Sequential(
          (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (1): BatchNorm2d(72, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          (2): ReLU(inplace=True)
          (3): Conv2d(72, 72, kernel_size=(3, 3), stride=(1, 1), pad
ding=(1, 1), groups=72, bias=False)
          (4): BatchNorm2d(72, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          (5): ReLU(inplace=True)
          (6): Conv2d(72, 24, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (7): BatchNorm2d(24, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
        )
      )
      (2): _InvertedResidual(
        (layers): Sequential(
          (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (1): BatchNorm2d(72, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          (2): ReLU(inplace=True)
          (3): Conv2d(72, 72, kernel_size=(3, 3), stride=(1, 1), pad

```

```

ding=(1, 1), groups=72, bias=False)
    (4): BatchNorm2d(72, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(72, 24, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
    (7): BatchNorm2d(24, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
  )
)
(9): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
      (1): BatchNorm2d(72, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(72, 72, kernel_size=(5, 5), stride=(2, 2), pad
ding=(2, 2), groups=72, bias=False)
      (4): BatchNorm2d(72, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(72, 40, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
      (7): BatchNorm2d(40, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(120, 120, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=120, bias=False)
      (4): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(120, 40, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(40, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (2): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(120, 120, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=120, bias=False)
      (4): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(120, 40, kernel_size=(1, 1), stride=(1, 1), bi

```



```

as=False)
    (7): BatchNorm2d(40, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
)
(10): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(40, 240, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(240, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(240, 240, kernel_size=(5, 5), stride=(2, 2), p
adding=(2, 2), groups=240, bias=False)
      (4): BatchNorm2d(240, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(240, 80, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(80, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(480, 480, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=480, bias=False)
      (4): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(480, 80, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(80, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (2): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(480, 480, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=480, bias=False)
      (4): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(480, 80, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(80, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
)

```

```

)
(11): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(480, 480, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=480, bias=False)
      (4): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(480, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (7): BatchNorm2d(96, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=576, bias=False)
      (4): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (7): BatchNorm2d(96, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
    )
  )
)
(12): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(576, 576, kernel_size=(5, 5), stride=(2, 2), padding=(2, 2), groups=576, bias=False)
      (4): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(576, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (7): BatchNorm2d(192, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),

```

```

bias=False)
    (1): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
    (4): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (7): BatchNorm2d(192, eps=1e-05, momentum=0.000299999999999
996696, affine=True, track_running_stats=True)
    )
    )
    (2): _InvertedResidual(
    (layers): Sequential(
    (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (1): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
    (4): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (7): BatchNorm2d(192, eps=1e-05, momentum=0.000299999999999
996696, affine=True, track_running_stats=True)
    )
    )
    (3): _InvertedResidual(
    (layers): Sequential(
    (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (1): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
    (4): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (7): BatchNorm2d(192, eps=1e-05, momentum=0.000299999999999
996696, affine=True, track_running_stats=True)
    )
    )
    )
    (13): Sequential(
    (0): _InvertedResidual(
    (layers): Sequential(
    (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (1): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(1152, 1152, kernel_size=(3, 3), stride=(1, 1),

```

```
padding=(1, 1), groups=1152, bias=False)
    (4): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(1152, 320, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (7): BatchNorm2d(320, eps=1e-05, momentum=0.000299999999999
996696, affine=True, track_running_stats=True)
    )
    )
    (14): Conv2d(320, 1280, kernel_size=(1, 1), stride=(1, 1), bias=
False)
    (15): BatchNorm2d(1280, eps=1e-05, momentum=0.0002999999999999966
96, affine=True, track_running_stats=True)
    (16): ReLU(inplace=True)
    )
    (classifier): Sequential(
      (0): Dropout(p=0.2, inplace=True)
      (1): Linear(in_features=1280, out_features=12, bias=True)
    )
  )
)
```

In [0]:

```
# criterion and optimzier
from torch import optim
criterion_mnasnet = nn.CrossEntropyLoss()
optimizer_mnasnet = optim.Adam(mnasnet.classifier.parameters(), lr=0.001)
```

In [136]:

```
mnasnet, train_loss_coll_mnasnet, valid_loss_coll_mnasnet = train(1, 50, np.Inf,  
data_loaders, mnasnet, optimizer_mnasnet, criterion_mnasnet, use_cuda, "./checkp  
oint/current_checkpoint_mnasnet.pt", "./best_model/best_model_mnasnet.pt")
```

```
/usr/local/lib/python3.6/dist-packages/PIL/Image.py:989: UserWarning: Palette images with Transparency expressed in bytes should be converted to RGBA images
  "Palette images with Transparency expressed in bytes should be "
```

```

Epoch: 1      Train Loss: 0.903680    Valid Loss: 0.347353    Trai
n Acc: 76.61   Valid Acc: 88.72
Validation loss decreased (inf --> 0.347353). Saving model ...
Epoch: 2      Train Loss: 0.495824    Valid Loss: 0.305611    Trai
n Acc: 85.48   Valid Acc: 88.72
Validation loss decreased (0.347353 --> 0.305611). Saving model ...
Epoch: 3      Train Loss: 0.434674    Valid Loss: 0.274993    Trai
n Acc: 86.58   Valid Acc: 91.34
Validation loss decreased (0.305611 --> 0.274993). Saving model ...
Epoch: 4      Train Loss: 0.386891    Valid Loss: 0.263900    Trai
n Acc: 87.52   Valid Acc: 91.04
Validation loss decreased (0.274993 --> 0.263900). Saving model ...
Epoch: 5      Train Loss: 0.384876    Valid Loss: 0.262156    Trai
n Acc: 87.50   Valid Acc: 91.65
Validation loss decreased (0.263900 --> 0.262156). Saving model ...
Epoch: 6      Train Loss: 0.362985    Valid Loss: 0.273837    Trai
n Acc: 88.01   Valid Acc: 90.57
Epoch: 7      Train Loss: 0.354499    Valid Loss: 0.256227    Trai
n Acc: 88.36   Valid Acc: 90.73
Validation loss decreased (0.262156 --> 0.256227). Saving model ...
Epoch: 8      Train Loss: 0.335547    Valid Loss: 0.270272    Trai
n Acc: 88.87   Valid Acc: 90.26
Epoch: 9      Train Loss: 0.342181    Valid Loss: 0.262449    Trai
n Acc: 88.54   Valid Acc: 91.19
Epoch: 10     Train Loss: 0.337691    Valid Loss: 0.274467    Trai
n Acc: 88.66   Valid Acc: 91.19
Epoch: 11     Train Loss: 0.327475    Valid Loss: 0.255171    Trai
n Acc: 89.35   Valid Acc: 91.50
Validation loss decreased (0.256227 --> 0.255171). Saving model ...
Epoch: 12     Train Loss: 0.324602    Valid Loss: 0.268045    Trai
n Acc: 89.24   Valid Acc: 91.19
Epoch: 13     Train Loss: 0.312370    Valid Loss: 0.265311    Trai
n Acc: 89.93   Valid Acc: 90.88
Epoch: 14     Train Loss: 0.311255    Valid Loss: 0.282081    Trai
n Acc: 89.49   Valid Acc: 90.26
Epoch: 15     Train Loss: 0.307964    Valid Loss: 0.271054    Trai
n Acc: 89.74   Valid Acc: 91.19
Epoch: 16     Train Loss: 0.309928    Valid Loss: 0.281529    Trai
n Acc: 89.60   Valid Acc: 89.95
Epoch: 17     Train Loss: 0.305262    Valid Loss: 0.275524    Trai
n Acc: 90.00   Valid Acc: 90.73
Epoch: 18     Train Loss: 0.300691    Valid Loss: 0.300486    Trai
n Acc: 89.81   Valid Acc: 90.26
Epoch: 19     Train Loss: 0.303079    Valid Loss: 0.297222    Trai
n Acc: 89.54   Valid Acc: 90.42
Epoch: 20     Train Loss: 0.307517    Valid Loss: 0.299297    Trai
n Acc: 89.51   Valid Acc: 90.26
Epoch: 21     Train Loss: 0.302569    Valid Loss: 0.298387    Trai
n Acc: 89.64   Valid Acc: 90.57
Epoch: 22     Train Loss: 0.297734    Valid Loss: 0.297566    Trai
n Acc: 89.69   Valid Acc: 90.88
Epoch: 23     Train Loss: 0.306691    Valid Loss: 0.300016    Trai
n Acc: 89.44   Valid Acc: 91.50
Epoch: 24     Train Loss: 0.284512    Valid Loss: 0.316611    Trai
n Acc: 90.31   Valid Acc: 90.73
Epoch: 25     Train Loss: 0.285716    Valid Loss: 0.302546    Trai
n Acc: 90.03   Valid Acc: 90.57
Epoch: 26     Train Loss: 0.291073    Valid Loss: 0.315948    Trai
n Acc: 90.17   Valid Acc: 90.11
Epoch: 27     Train Loss: 0.289049    Valid Loss: 0.298488    Trai
n Acc: 90.29   Valid Acc: 90.73

```

Epoch: 28	Train Loss: 0.282625	Valid Loss: 0.307370	Trai
n Acc: 90.67	Valid Acc: 90.42		
Epoch: 29	Train Loss: 0.274060	Valid Loss: 0.310396	Trai
n Acc: 90.65	Valid Acc: 90.42		
Epoch: 30	Train Loss: 0.292918	Valid Loss: 0.294143	Trai
n Acc: 90.07	Valid Acc: 90.73		
Epoch: 31	Train Loss: 0.288762	Valid Loss: 0.317154	Trai
n Acc: 90.17	Valid Acc: 89.80		
Epoch: 32	Train Loss: 0.283586	Valid Loss: 0.312044	Trai
n Acc: 90.28	Valid Acc: 90.42		
Epoch: 33	Train Loss: 0.292866	Valid Loss: 0.301301	Trai
n Acc: 90.21	Valid Acc: 89.80		
Epoch: 34	Train Loss: 0.291712	Valid Loss: 0.301402	Trai
n Acc: 89.94	Valid Acc: 90.11		
Epoch: 35	Train Loss: 0.294574	Valid Loss: 0.315814	Trai
n Acc: 90.25	Valid Acc: 90.11		
Epoch: 36	Train Loss: 0.282532	Valid Loss: 0.321962	Trai
n Acc: 90.19	Valid Acc: 89.80		
Epoch: 37	Train Loss: 0.293987	Valid Loss: 0.312351	Trai
n Acc: 90.11	Valid Acc: 89.49		
Epoch: 38	Train Loss: 0.293287	Valid Loss: 0.306789	Trai
n Acc: 90.19	Valid Acc: 89.64		
Epoch: 39	Train Loss: 0.287010	Valid Loss: 0.328278	Trai
n Acc: 90.43	Valid Acc: 88.87		
Epoch: 40	Train Loss: 0.256629	Valid Loss: 0.322630	Trai
n Acc: 91.34	Valid Acc: 89.64		
Epoch: 41	Train Loss: 0.288621	Valid Loss: 0.331932	Trai
n Acc: 90.04	Valid Acc: 89.03		
Epoch: 42	Train Loss: 0.288586	Valid Loss: 0.316736	Trai
n Acc: 90.13	Valid Acc: 89.95		
Epoch: 43	Train Loss: 0.278051	Valid Loss: 0.331543	Trai
n Acc: 90.67	Valid Acc: 89.80		
Epoch: 44	Train Loss: 0.273728	Valid Loss: 0.327050	Trai
n Acc: 90.82	Valid Acc: 89.64		
Epoch: 45	Train Loss: 0.296365	Valid Loss: 0.325935	Trai
n Acc: 89.83	Valid Acc: 89.49		
Epoch: 46	Train Loss: 0.287438	Valid Loss: 0.317392	Trai
n Acc: 90.25	Valid Acc: 89.80		
Epoch: 47	Train Loss: 0.278296	Valid Loss: 0.326502	Trai
n Acc: 90.28	Valid Acc: 89.95		
Epoch: 48	Train Loss: 0.273117	Valid Loss: 0.335560	Trai
n Acc: 90.53	Valid Acc: 89.95		
Epoch: 49	Train Loss: 0.298702	Valid Loss: 0.322030	Trai
n Acc: 89.96	Valid Acc: 89.18		
Epoch: 50	Train Loss: 0.271728	Valid Loss: 0.317684	Trai
n Acc: 90.51	Valid Acc: 89.80		

Serialized MNASET model for deployment

In [0]:

```
mnaset_loaded_model = models.mnasnet1_0(pretrained=True)
```


In [0]:

```
print(mnaset_loaded_model)
```

```

MNASNet(
  (layers): Sequential(
    (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(2, 2), padding=
(1, 1), bias=False)
    (1): BatchNorm2d(32, eps=1e-05, momentum=0.00029999999999999696,
affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), groups=32, bias=False)
    (4): BatchNorm2d(32, eps=1e-05, momentum=0.00029999999999999696,
affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(32, 16, kernel_size=(1, 1), stride=(1, 1), bias=Fals
e)
    (7): BatchNorm2d(16, eps=1e-05, momentum=0.00029999999999999696,
affine=True, track_running_stats=True)
    (8): Sequential(
      (0): _InvertedResidual(
        (layers): Sequential(
          (0): Conv2d(16, 48, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (1): BatchNorm2d(48, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          (2): ReLU(inplace=True)
          (3): Conv2d(48, 48, kernel_size=(3, 3), stride=(2, 2), pad
ding=(1, 1), groups=48, bias=False)
          (4): BatchNorm2d(48, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          (5): ReLU(inplace=True)
          (6): Conv2d(48, 24, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (7): BatchNorm2d(24, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
        )
      )
      (1): _InvertedResidual(
        (layers): Sequential(
          (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (1): BatchNorm2d(72, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          (2): ReLU(inplace=True)
          (3): Conv2d(72, 72, kernel_size=(3, 3), stride=(1, 1), pad
ding=(1, 1), groups=72, bias=False)
          (4): BatchNorm2d(72, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          (5): ReLU(inplace=True)
          (6): Conv2d(72, 24, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (7): BatchNorm2d(24, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
        )
      )
      (2): _InvertedResidual(
        (layers): Sequential(
          (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (1): BatchNorm2d(72, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          (2): ReLU(inplace=True)
          (3): Conv2d(72, 72, kernel_size=(3, 3), stride=(1, 1), pad

```

```

ding=(1, 1), groups=72, bias=False)
    (4): BatchNorm2d(72, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(72, 24, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
    (7): BatchNorm2d(24, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
  )
)
(9): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
      (1): BatchNorm2d(72, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(72, 72, kernel_size=(5, 5), stride=(2, 2), pad
ding=(2, 2), groups=72, bias=False)
      (4): BatchNorm2d(72, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(72, 40, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
      (7): BatchNorm2d(40, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(120, 120, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=120, bias=False)
      (4): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(120, 40, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(40, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (2): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(120, 120, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=120, bias=False)
      (4): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(120, 40, kernel_size=(1, 1), stride=(1, 1), bi

```

```

as=False)
    (7): BatchNorm2d(40, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
)
(10): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(40, 240, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(240, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(240, 240, kernel_size=(5, 5), stride=(2, 2), p
adding=(2, 2), groups=240, bias=False)
      (4): BatchNorm2d(240, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(240, 80, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(80, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(480, 480, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=480, bias=False)
      (4): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(480, 80, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(80, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (2): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(480, 480, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=480, bias=False)
      (4): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(480, 80, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(80, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
)

```

```

)
(11): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(480, 480, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=480, bias=False)
      (4): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(480, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (7): BatchNorm2d(96, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=576, bias=False)
      (4): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (7): BatchNorm2d(96, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
    )
  )
)
(12): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(576, 576, kernel_size=(5, 5), stride=(2, 2), padding=(2, 2), groups=576, bias=False)
      (4): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(576, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (7): BatchNorm2d(192, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),

```

```

bias=False)
    (1): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
    (4): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (7): BatchNorm2d(192, eps=1e-05, momentum=0.000299999999999
996696, affine=True, track_running_stats=True)
  )
)
(2): _InvertedResidual(
(layers): Sequential(
(0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
(1): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
(2): ReLU(inplace=True)
(3): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
(4): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
(5): ReLU(inplace=True)
(6): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
(7): BatchNorm2d(192, eps=1e-05, momentum=0.000299999999999
996696, affine=True, track_running_stats=True)
)
)
(3): _InvertedResidual(
(layers): Sequential(
(0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
(1): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
(2): ReLU(inplace=True)
(3): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
(4): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
(5): ReLU(inplace=True)
(6): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
(7): BatchNorm2d(192, eps=1e-05, momentum=0.000299999999999
996696, affine=True, track_running_stats=True)
)
)
)
(13): Sequential(
(0): _InvertedResidual(
(layers): Sequential(
(0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
(1): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
(2): ReLU(inplace=True)
(3): Conv2d(1152, 1152, kernel_size=(3, 3), stride=(1, 1),

```

```
padding=(1, 1), groups=1152, bias=False)
    (4): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(1152, 320, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (7): BatchNorm2d(320, eps=1e-05, momentum=0.000299999999999
996696, affine=True, track_running_stats=True)
    )
    )
    (14): Conv2d(320, 1280, kernel_size=(1, 1), stride=(1, 1), bias=
False)
    (15): BatchNorm2d(1280, eps=1e-05, momentum=0.0002999999999999966
96, affine=True, track_running_stats=True)
    (16): ReLU(inplace=True)
    )
    (classifier): Sequential(
      (0): Dropout(p=0.2, inplace=True)
      (1): Linear(in_features=1280, out_features=1000, bias=True)
    )
  )
)
```

In [0]:

```
n_inputs = mnaset_loaded_model.classifier[1].in_features
n_classes = len(class_names)
mnaset_loaded_model.classifier = nn.Sequential(
    nn.Dropout(0.2, inplace = True),
    nn.Linear(n_inputs, n_classes, bias = True))
```

In [0]:

```
print(mnaset_loaded_model)
```



```

MNASNet(
  (layers): Sequential(
    (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(2, 2), padding=
(1, 1), bias=False)
    (1): BatchNorm2d(32, eps=1e-05, momentum=0.00029999999999999696,
affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), groups=32, bias=False)
    (4): BatchNorm2d(32, eps=1e-05, momentum=0.00029999999999999696,
affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(32, 16, kernel_size=(1, 1), stride=(1, 1), bias=Fals
e)
    (7): BatchNorm2d(16, eps=1e-05, momentum=0.00029999999999999696,
affine=True, track_running_stats=True)
    (8): Sequential(
      (0): _InvertedResidual(
        (layers): Sequential(
          (0): Conv2d(16, 48, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (1): BatchNorm2d(48, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          (2): ReLU(inplace=True)
          (3): Conv2d(48, 48, kernel_size=(3, 3), stride=(2, 2), pad
ding=(1, 1), groups=48, bias=False)
          (4): BatchNorm2d(48, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          (5): ReLU(inplace=True)
          (6): Conv2d(48, 24, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (7): BatchNorm2d(24, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
        )
      )
      (1): _InvertedResidual(
        (layers): Sequential(
          (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (1): BatchNorm2d(72, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          (2): ReLU(inplace=True)
          (3): Conv2d(72, 72, kernel_size=(3, 3), stride=(1, 1), pad
ding=(1, 1), groups=72, bias=False)
          (4): BatchNorm2d(72, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          (5): ReLU(inplace=True)
          (6): Conv2d(72, 24, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (7): BatchNorm2d(24, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
        )
      )
      (2): _InvertedResidual(
        (layers): Sequential(
          (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (1): BatchNorm2d(72, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          (2): ReLU(inplace=True)
          (3): Conv2d(72, 72, kernel_size=(3, 3), stride=(1, 1), pad

```

```

ding=(1, 1), groups=72, bias=False)
    (4): BatchNorm2d(72, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(72, 24, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
    (7): BatchNorm2d(24, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
  )
)
(9): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
      (1): BatchNorm2d(72, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(72, 72, kernel_size=(5, 5), stride=(2, 2), pad
ding=(2, 2), groups=72, bias=False)
      (4): BatchNorm2d(72, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(72, 40, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
      (7): BatchNorm2d(40, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(120, 120, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=120, bias=False)
      (4): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(120, 40, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(40, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (2): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(120, 120, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=120, bias=False)
      (4): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(120, 40, kernel_size=(1, 1), stride=(1, 1), bi

```

```
as=False)
    (7): BatchNorm2d(40, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
)
(10): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(40, 240, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(240, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(240, 240, kernel_size=(5, 5), stride=(2, 2), p
adding=(2, 2), groups=240, bias=False)
      (4): BatchNorm2d(240, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(240, 80, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(80, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(480, 480, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=480, bias=False)
      (4): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(480, 80, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(80, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (2): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(480, 480, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=480, bias=False)
      (4): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(480, 80, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(80, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
)
```

```

)
(11): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(480, 480, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=480, bias=False)
      (4): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(480, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (7): BatchNorm2d(96, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=576, bias=False)
      (4): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (7): BatchNorm2d(96, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
    )
  )
)
(12): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(576, 576, kernel_size=(5, 5), stride=(2, 2), padding=(2, 2), groups=576, bias=False)
      (4): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(576, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (7): BatchNorm2d(192, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),

```

```

bias=False)
    (1): BatchNorm2d(1152, eps=1e-05, momentum=0.00029999999999
9996696, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
    (4): BatchNorm2d(1152, eps=1e-05, momentum=0.00029999999999
9996696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (7): BatchNorm2d(192, eps=1e-05, momentum=0.00029999999999
996696, affine=True, track_running_stats=True)
    )
    )
    (2): _InvertedResidual(
(layers): Sequential(
  (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
  (1): BatchNorm2d(1152, eps=1e-05, momentum=0.00029999999999
9996696, affine=True, track_running_stats=True)
  (2): ReLU(inplace=True)
  (3): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
  (4): BatchNorm2d(1152, eps=1e-05, momentum=0.00029999999999
9996696, affine=True, track_running_stats=True)
  (5): ReLU(inplace=True)
  (6): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
  (7): BatchNorm2d(192, eps=1e-05, momentum=0.00029999999999
996696, affine=True, track_running_stats=True)
  )
  )
    (3): _InvertedResidual(
(layers): Sequential(
  (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
  (1): BatchNorm2d(1152, eps=1e-05, momentum=0.00029999999999
9996696, affine=True, track_running_stats=True)
  (2): ReLU(inplace=True)
  (3): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
  (4): BatchNorm2d(1152, eps=1e-05, momentum=0.00029999999999
9996696, affine=True, track_running_stats=True)
  (5): ReLU(inplace=True)
  (6): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
  (7): BatchNorm2d(192, eps=1e-05, momentum=0.00029999999999
996696, affine=True, track_running_stats=True)
  )
  )
    )
    (13): Sequential(
  (0): _InvertedResidual(
(layers): Sequential(
  (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
  (1): BatchNorm2d(1152, eps=1e-05, momentum=0.00029999999999
9996696, affine=True, track_running_stats=True)
  (2): ReLU(inplace=True)
  (3): Conv2d(1152, 1152, kernel_size=(3, 3), stride=(1, 1),

```

```
padding=(1, 1), groups=1152, bias=False)
    (4): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(1152, 320, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (7): BatchNorm2d(320, eps=1e-05, momentum=0.000299999999999
996696, affine=True, track_running_stats=True)
    )
    )
    (14): Conv2d(320, 1280, kernel_size=(1, 1), stride=(1, 1), bias=
False)
    (15): BatchNorm2d(1280, eps=1e-05, momentum=0.0002999999999999966
96, affine=True, track_running_stats=True)
    (16): ReLU(inplace=True)
    )
    (classifier): Sequential(
      (0): Dropout(p=0.2, inplace=True)
      (1): Linear(in_features=1280, out_features=12, bias=True)
    )
  )
)
```

In [0]:

```
# load state of saved best model
ckp_path = './best_model/best_model_mnasnet.pt'
##checkpoint = torch.load(ckp_path, map_location=torch.device('cpu'))
# initialize state_dict from checkpoint to model
#mnaset_loaded_model.load_state_dict(checkpoint['state_dict'])
```

Out[0]:

<All keys matched successfully>

In [0]:

```
print(mnaset_loaded_model)
```

```

MNASNet(
  (layers): Sequential(
    (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(2, 2), padding=
(1, 1), bias=False)
    (1): BatchNorm2d(32, eps=1e-05, momentum=0.00029999999999999696,
affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), groups=32, bias=False)
    (4): BatchNorm2d(32, eps=1e-05, momentum=0.00029999999999999696,
affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(32, 16, kernel_size=(1, 1), stride=(1, 1), bias=Fals
e)
    (7): BatchNorm2d(16, eps=1e-05, momentum=0.00029999999999999696,
affine=True, track_running_stats=True)
    (8): Sequential(
      (0): _InvertedResidual(
        (layers): Sequential(
          (0): Conv2d(16, 48, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (1): BatchNorm2d(48, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          (2): ReLU(inplace=True)
          (3): Conv2d(48, 48, kernel_size=(3, 3), stride=(2, 2), pad
ding=(1, 1), groups=48, bias=False)
          (4): BatchNorm2d(48, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          (5): ReLU(inplace=True)
          (6): Conv2d(48, 24, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (7): BatchNorm2d(24, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
        )
      )
      (1): _InvertedResidual(
        (layers): Sequential(
          (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (1): BatchNorm2d(72, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          (2): ReLU(inplace=True)
          (3): Conv2d(72, 72, kernel_size=(3, 3), stride=(1, 1), pad
ding=(1, 1), groups=72, bias=False)
          (4): BatchNorm2d(72, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          (5): ReLU(inplace=True)
          (6): Conv2d(72, 24, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (7): BatchNorm2d(24, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
        )
      )
      (2): _InvertedResidual(
        (layers): Sequential(
          (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
          (1): BatchNorm2d(72, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          (2): ReLU(inplace=True)
          (3): Conv2d(72, 72, kernel_size=(3, 3), stride=(1, 1), pad

```



```

ding=(1, 1), groups=72, bias=False)
    (4): BatchNorm2d(72, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(72, 24, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
    (7): BatchNorm2d(24, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
  )
)
(9): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
      (1): BatchNorm2d(72, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(72, 72, kernel_size=(5, 5), stride=(2, 2), pad
ding=(2, 2), groups=72, bias=False)
      (4): BatchNorm2d(72, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(72, 40, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
      (7): BatchNorm2d(40, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(120, 120, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=120, bias=False)
      (4): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(120, 40, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(40, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (2): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(120, 120, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=120, bias=False)
      (4): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(120, 40, kernel_size=(1, 1), stride=(1, 1), bi

```

```
as=False)
    (7): BatchNorm2d(40, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
)
(10): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(40, 240, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(240, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(240, 240, kernel_size=(5, 5), stride=(2, 2), p
adding=(2, 2), groups=240, bias=False)
      (4): BatchNorm2d(240, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(240, 80, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(80, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(480, 480, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=480, bias=False)
      (4): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(480, 80, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(80, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (2): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(480, 480, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=480, bias=False)
      (4): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(480, 80, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(80, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
)
```

```

)
(11): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(480, 480, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=480, bias=False)
      (4): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(480, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (7): BatchNorm2d(96, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=576, bias=False)
      (4): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (7): BatchNorm2d(96, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
    )
  )
)
(12): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(576, 576, kernel_size=(5, 5), stride=(2, 2), padding=(2, 2), groups=576, bias=False)
      (4): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(576, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (7): BatchNorm2d(192, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),

```

```

bias=False)
    (1): BatchNorm2d(1152, eps=1e-05, momentum=0.00029999999999
9996696, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
    (4): BatchNorm2d(1152, eps=1e-05, momentum=0.00029999999999
9996696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (7): BatchNorm2d(192, eps=1e-05, momentum=0.000299999999999
996696, affine=True, track_running_stats=True)
    )
    )
    (2): _InvertedResidual(
    (layers): Sequential(
    (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (1): BatchNorm2d(1152, eps=1e-05, momentum=0.00029999999999
9996696, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
    (4): BatchNorm2d(1152, eps=1e-05, momentum=0.00029999999999
9996696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (7): BatchNorm2d(192, eps=1e-05, momentum=0.000299999999999
996696, affine=True, track_running_stats=True)
    )
    )
    (3): _InvertedResidual(
    (layers): Sequential(
    (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (1): BatchNorm2d(1152, eps=1e-05, momentum=0.00029999999999
9996696, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
    (4): BatchNorm2d(1152, eps=1e-05, momentum=0.00029999999999
9996696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (7): BatchNorm2d(192, eps=1e-05, momentum=0.000299999999999
996696, affine=True, track_running_stats=True)
    )
    )
    )
    (13): Sequential(
    (0): _InvertedResidual(
    (layers): Sequential(
    (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (1): BatchNorm2d(1152, eps=1e-05, momentum=0.00029999999999
9996696, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(1152, 1152, kernel_size=(3, 3), stride=(1, 1),

```

```
padding=(1, 1), groups=1152, bias=False)
    (4): BatchNorm2d(1152, eps=1e-05, momentum=0.000299999999999
9996696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(1152, 320, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (7): BatchNorm2d(320, eps=1e-05, momentum=0.000299999999999
996696, affine=True, track_running_stats=True)
    )
    )
    (14): Conv2d(320, 1280, kernel_size=(1, 1), stride=(1, 1), bias=
False)
    (15): BatchNorm2d(1280, eps=1e-05, momentum=0.000299999999999996
96, affine=True, track_running_stats=True)
    (16): ReLU(inplace=True)
    )
    (classifier): Sequential(
      (0): Dropout(p=0.2, inplace=True)
      (1): Linear(in_features=1280, out_features=12, bias=True)
    )
  )
)
```

In [0]:

```
# load state of saved best checkpoint
ckp_path = './best_model/best_model_mnasnet.pt'
best_model_loaded, optimizer_best, start_epoch, valid_loss_min, train_loss_coll_
best, valid_loss_coll_best = load_ckpt(ckp_path, mnasnet, optimizer_mnasnet)
```

In [0]:

```
mnaset_loaded_model.eval()
example = torch.rand(1, 3, 224, 224)
if use_cuda:
    example = example.cuda()
traced_script_module = torch.jit.trace(best_model_loaded, example)
traced_script_module.save("./serialized_model/serialized_mnaset_model_v2.pt")
```

In [0]:

```
%pwd
```

Out[0]:

```
'/content/drive/My Drive/MyDrive/Udacity/CNN Chinese Zodiac Signs'
```

In [0]:

```
# load state of saved best model
ckp_path = './best_model/best_model_mnasnet.pt'
model_mnasnet, optimizer_mnasnet, start_epoch, valid_loss_min, train_loss_coll_m
nasnet, valid_loss_coll_mnasnet = load_ckpt(ckp_path, mnasnet, optimizer_mnasnet)
```

In [127]:

```
print("model = ", model_mnasnet)
print("optimizer = ", optimizer_mnasnet)
print("start_epoch = ", start_epoch)
print("valid_loss_min = ", valid_loss_min)
print("valid_loss_min = {:.6f}".format(valid_loss_min))
print("train_loss_coll_mnasnet = ", train_loss_coll_mnasnet)
print("valid_loss_coll_mnasnet = ", valid_loss_coll_mnasnet)
```

```

model = MNASNet(
    (layers): Sequential(
      (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(2, 2), padding=
(1, 1), bias=False)
      (1): BatchNorm2d(32, eps=1e-05, momentum=0.00029999999999999696,
affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1), groups=32, bias=False)
      (4): BatchNorm2d(32, eps=1e-05, momentum=0.00029999999999999696,
affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(32, 16, kernel_size=(1, 1), stride=(1, 1), bias=Fals
e)
      (7): BatchNorm2d(16, eps=1e-05, momentum=0.00029999999999999696,
affine=True, track_running_stats=True)
      (8): Sequential(
        (0): _InvertedResidual(
          (layers): Sequential(
            (0): Conv2d(16, 48, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
            (1): BatchNorm2d(48, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
            (2): ReLU(inplace=True)
            (3): Conv2d(48, 48, kernel_size=(3, 3), stride=(2, 2), pad
ding=(1, 1), groups=48, bias=False)
            (4): BatchNorm2d(48, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
            (5): ReLU(inplace=True)
            (6): Conv2d(48, 24, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
            (7): BatchNorm2d(24, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          )
        )
        (1): _InvertedResidual(
          (layers): Sequential(
            (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
            (1): BatchNorm2d(72, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
            (2): ReLU(inplace=True)
            (3): Conv2d(72, 72, kernel_size=(3, 3), stride=(1, 1), pad
ding=(1, 1), groups=72, bias=False)
            (4): BatchNorm2d(72, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
            (5): ReLU(inplace=True)
            (6): Conv2d(72, 24, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
            (7): BatchNorm2d(24, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
          )
        )
        (2): _InvertedResidual(
          (layers): Sequential(
            (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
            (1): BatchNorm2d(72, eps=1e-05, momentum=0.000299999999999
9696, affine=True, track_running_stats=True)
            (2): ReLU(inplace=True)
            (3): Conv2d(72, 72, kernel_size=(3, 3), stride=(1, 1), pad

```

```

ding=(1, 1), groups=72, bias=False)
    (4): BatchNorm2d(72, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(72, 24, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
    (7): BatchNorm2d(24, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
  )
)
(9): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
      (1): BatchNorm2d(72, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(72, 72, kernel_size=(5, 5), stride=(2, 2), pad
ding=(2, 2), groups=72, bias=False)
      (4): BatchNorm2d(72, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(72, 40, kernel_size=(1, 1), stride=(1, 1), bia
s=False)
      (7): BatchNorm2d(40, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(120, 120, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=120, bias=False)
      (4): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(120, 40, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(40, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (2): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(120, 120, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=120, bias=False)
      (4): BatchNorm2d(120, eps=1e-05, momentum=0.00029999999999999
996696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(120, 40, kernel_size=(1, 1), stride=(1, 1), bi

```



```
as=False)
    (7): BatchNorm2d(40, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
)
(10): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(40, 240, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(240, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(240, 240, kernel_size=(5, 5), stride=(2, 2), p
adding=(2, 2), groups=240, bias=False)
      (4): BatchNorm2d(240, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(240, 80, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(80, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(480, 480, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=480, bias=False)
      (4): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(480, 80, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(80, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
  (2): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(480, 480, kernel_size=(5, 5), stride=(1, 1), p
adding=(2, 2), groups=480, bias=False)
      (4): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(480, 80, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
      (7): BatchNorm2d(80, eps=1e-05, momentum=0.00029999999999999
96696, affine=True, track_running_stats=True)
    )
  )
)
```

```

)
(11): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(480, 480, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=480, bias=False)
      (4): BatchNorm2d(480, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(480, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (7): BatchNorm2d(96, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(576, 576, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=576, bias=False)
      (4): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (7): BatchNorm2d(96, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
    )
  )
)
(12): Sequential(
  (0): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (2): ReLU(inplace=True)
      (3): Conv2d(576, 576, kernel_size=(5, 5), stride=(2, 2), padding=(2, 2), groups=576, bias=False)
      (4): BatchNorm2d(576, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
      (5): ReLU(inplace=True)
      (6): Conv2d(576, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (7): BatchNorm2d(192, eps=1e-05, momentum=0.00029999999999999996, affine=True, track_running_stats=True)
    )
  )
  (1): _InvertedResidual(
    (layers): Sequential(
      (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),

```

```

bias=False)
    (1): BatchNorm2d(1152, eps=1e-05, momentum=0.00029999999999
9996696, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
    (4): BatchNorm2d(1152, eps=1e-05, momentum=0.00029999999999
9996696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (7): BatchNorm2d(192, eps=1e-05, momentum=0.00029999999999
996696, affine=True, track_running_stats=True)
  )
)
(2): _InvertedResidual(
(layers): Sequential(
(0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
(1): BatchNorm2d(1152, eps=1e-05, momentum=0.00029999999999
9996696, affine=True, track_running_stats=True)
(2): ReLU(inplace=True)
(3): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
(4): BatchNorm2d(1152, eps=1e-05, momentum=0.00029999999999
9996696, affine=True, track_running_stats=True)
(5): ReLU(inplace=True)
(6): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
(7): BatchNorm2d(192, eps=1e-05, momentum=0.00029999999999
996696, affine=True, track_running_stats=True)
)
)
(3): _InvertedResidual(
(layers): Sequential(
(0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
(1): BatchNorm2d(1152, eps=1e-05, momentum=0.00029999999999
9996696, affine=True, track_running_stats=True)
(2): ReLU(inplace=True)
(3): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
(4): BatchNorm2d(1152, eps=1e-05, momentum=0.00029999999999
9996696, affine=True, track_running_stats=True)
(5): ReLU(inplace=True)
(6): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
(7): BatchNorm2d(192, eps=1e-05, momentum=0.00029999999999
996696, affine=True, track_running_stats=True)
)
)
)
(13): Sequential(
(0): _InvertedResidual(
(layers): Sequential(
(0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
(1): BatchNorm2d(1152, eps=1e-05, momentum=0.00029999999999
9996696, affine=True, track_running_stats=True)
(2): ReLU(inplace=True)
(3): Conv2d(1152, 1152, kernel_size=(3, 3), stride=(1, 1),

```

```
padding=(1, 1), groups=1152, bias=False)
    (4): BatchNorm2d(1152, eps=1e-05, momentum=0.0002999999999
9996696, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): Conv2d(1152, 320, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (7): BatchNorm2d(320, eps=1e-05, momentum=0.0002999999999
996696, affine=True, track_running_stats=True)
    )
    )
    (14): Conv2d(320, 1280, kernel_size=(1, 1), stride=(1, 1), bias=
False)
    (15): BatchNorm2d(1280, eps=1e-05, momentum=0.000299999999999996
96, affine=True, track_running_stats=True)
    (16): ReLU(inplace=True)
    )
    (classifier): Sequential(
      (0): Dropout(p=0.2, inplace=True)
      (1): Linear(in_features=1280, out_features=12, bias=True)
    )
  )
optimizer = Adam (
Parameter Group 0
  amsgrad: False
  betas: (0.9, 0.999)
  eps: 1e-08
  lr: 0.001
  weight_decay: 0
)
start_epoch = 8
valid_loss_min = 0.2625579375779426
valid_loss_min = 0.262558
train_loss_coll_mnasnet = [0.9012736506521785, 0.49353406376235176,
0.4400406574548037, 0.40547131673434184, 0.3726654216460625, 0.36931
750062277985, 0.3568874763710857]
valid_loss_coll_mnasnet = [0.37046795724164205, 0.2873561561245822
6, 0.29507351071333404, 0.2955928564797137, 0.27394712678339606, 0.2
843822954910616, 0.2625579375779426]
```

Model DenseNet

In [24]:

```
densenet = models.densenet161(pretrained=True)
```

Downloading: "https://download.pytorch.org/models/densenet161-8d451a50.pth" to /root/.cache/torch/checkpoints/densenet161-8d451a50.pth
100%|██████████| 110M/110M [00:06<00:00, 19.1MB/s]

In [0]:

```
print(densenet)
```

```

DenseNet(
  (features): Sequential(
    (conv0): Conv2d(3, 96, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
    (norm0): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu0): ReLU(inplace=True)
    (pool0): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
    (denseblock1): _DenseBlock(
      (denselayer1): _DenseLayer(
        (norm1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(96, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      )
      (denselayer2): _DenseLayer(
        (norm1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(144, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      )
      (denselayer3): _DenseLayer(
        (norm1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(192, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      )
      (denselayer4): _DenseLayer(
        (norm1): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(240, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      )
      (denselayer5): _DenseLayer(
        (norm1): BatchNorm2d(288, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)

```

```

        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(288, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer6): _DenseLayer(
        (norm1): BatchNorm2d(336, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(336, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
)
(transition1): _Transition(
    (norm): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (relu): ReLU(inplace=True)
    (conv): Conv2d(384, 192, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
    (pool): AvgPool2d(kernel_size=2, stride=2, padding=0)
)
(denseblock2): _DenseBlock(
    (denselayer1): _DenseLayer(
        (norm1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(192, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer2): _DenseLayer(
        (norm1): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(240, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer3): _DenseLayer(
        (norm1): BatchNorm2d(288, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(288, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)

```

```

        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer4): _DenseLayer(
        (norm1): BatchNorm2d(336, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(336, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer5): _DenseLayer(
        (norm1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(384, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer6): _DenseLayer(
        (norm1): BatchNorm2d(432, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(432, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer7): _DenseLayer(
        (norm1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(480, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer8): _DenseLayer(
        (norm1): BatchNorm2d(528, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(528, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True

```



```

ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer9): _DenseLayer(
    (norm1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(576, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer10): _DenseLayer(
    (norm1): BatchNorm2d(624, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(624, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer11): _DenseLayer(
    (norm1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(672, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer12): _DenseLayer(
    (norm1): BatchNorm2d(720, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(720, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    )
    (transition2): _Transition(
    (norm): BatchNorm2d(768, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (relu): ReLU(inplace=True)
    (conv): Conv2d(768, 384, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
    (pool): AvgPool2d(kernel_size=2, stride=2, padding=0)

```

```

)
(denseblock3): _DenseBlock(
  (denselayer1): _DenseLayer(
    (norm1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(384, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  )
  (denselayer2): _DenseLayer(
    (norm1): BatchNorm2d(432, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(432, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  )
  (denselayer3): _DenseLayer(
    (norm1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(480, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  )
  (denselayer4): _DenseLayer(
    (norm1): BatchNorm2d(528, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(528, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  )
  (denselayer5): _DenseLayer(
    (norm1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(576, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  )

```

```

)
(denselayer6): _DenseLayer(
  (norm1): BatchNorm2d(624, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(624, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer7): _DenseLayer(
  (norm1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(672, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer8): _DenseLayer(
  (norm1): BatchNorm2d(720, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(720, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer9): _DenseLayer(
  (norm1): BatchNorm2d(768, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(768, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer10): _DenseLayer(
  (norm1): BatchNorm2d(816, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(816, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)

```

```

(denselayer11): _DenseLayer(
  (norm1): BatchNorm2d(864, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(864, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer12): _DenseLayer(
  (norm1): BatchNorm2d(912, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(912, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer13): _DenseLayer(
  (norm1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(960, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer14): _DenseLayer(
  (norm1): BatchNorm2d(1008, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(1008, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer15): _DenseLayer(
  (norm1): BatchNorm2d(1056, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(1056, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer16): _DenseLayer(

```

141/212

```
rue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1344, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer22): _DenseLayer(
    (norm1): BatchNorm2d(1392, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1392, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer23): _DenseLayer(
    (norm1): BatchNorm2d(1440, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1440, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer24): _DenseLayer(
    (norm1): BatchNorm2d(1488, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1488, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer25): _DenseLayer(
    (norm1): BatchNorm2d(1536, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1536, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer26): _DenseLayer(
    (norm1): BatchNorm2d(1584, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
```

```

        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1584, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer27): _DenseLayer(
        (norm1): BatchNorm2d(1632, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1632, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer28): _DenseLayer(
        (norm1): BatchNorm2d(1680, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1680, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer29): _DenseLayer(
        (norm1): BatchNorm2d(1728, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1728, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer30): _DenseLayer(
        (norm1): BatchNorm2d(1776, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1776, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer31): _DenseLayer(
        (norm1): BatchNorm2d(1824, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)

```

```

        (conv1): Conv2d(1824, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer32): _DenseLayer(
        (norm1): BatchNorm2d(1872, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1872, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer33): _DenseLayer(
        (norm1): BatchNorm2d(1920, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1920, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer34): _DenseLayer(
        (norm1): BatchNorm2d(1968, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1968, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer35): _DenseLayer(
        (norm1): BatchNorm2d(2016, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(2016, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer36): _DenseLayer(
        (norm1): BatchNorm2d(2064, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(2064, 192, kernel_size=(1, 1), stride=(1,

```



```

1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    )
    (transition3): _Transition(
    (norm): BatchNorm2d(2112, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace=True)
    (conv): Conv2d(2112, 1056, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (pool): AvgPool2d(kernel_size=2, stride=2, padding=0)
    )
    (denseblock4): _DenseBlock(
    (denselayer1): _DenseLayer(
    (norm1): BatchNorm2d(1056, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1056, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer2): _DenseLayer(
    (norm1): BatchNorm2d(1104, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1104, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer3): _DenseLayer(
    (norm1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer4): _DenseLayer(
    (norm1): BatchNorm2d(1200, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1200, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)

```

```

        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer5): _DenseLayer(
        (norm1): BatchNorm2d(1248, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1248, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer6): _DenseLayer(
        (norm1): BatchNorm2d(1296, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1296, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer7): _DenseLayer(
        (norm1): BatchNorm2d(1344, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1344, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer8): _DenseLayer(
        (norm1): BatchNorm2d(1392, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1392, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer9): _DenseLayer(
        (norm1): BatchNorm2d(1440, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1440, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)

```

```

        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer10): _DenseLayer(
        (norm1): BatchNorm2d(1488, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1488, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer11): _DenseLayer(
        (norm1): BatchNorm2d(1536, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1536, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer12): _DenseLayer(
        (norm1): BatchNorm2d(1584, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1584, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer13): _DenseLayer(
        (norm1): BatchNorm2d(1632, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1632, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer14): _DenseLayer(
        (norm1): BatchNorm2d(1680, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1680, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),

```

```

padding=(1, 1), bias=False)
    )
    (denselayer15): _DenseLayer(
        (norm1): BatchNorm2d(1728, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1728, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer16): _DenseLayer(
        (norm1): BatchNorm2d(1776, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1776, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer17): _DenseLayer(
        (norm1): BatchNorm2d(1824, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1824, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer18): _DenseLayer(
        (norm1): BatchNorm2d(1872, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1872, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer19): _DenseLayer(
        (norm1): BatchNorm2d(1920, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1920, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )

```

```

)
(denselayer20): _DenseLayer(
  (norm1): BatchNorm2d(1968, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(1968, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer21): _DenseLayer(
  (norm1): BatchNorm2d(2016, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(2016, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer22): _DenseLayer(
  (norm1): BatchNorm2d(2064, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(2064, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer23): _DenseLayer(
  (norm1): BatchNorm2d(2112, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(2112, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer24): _DenseLayer(
  (norm1): BatchNorm2d(2160, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(2160, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)

```

```
)  
(norm5): BatchNorm2d(2208, eps=1e-05, momentum=0.1, affine=True,  
track_running_stats=True)  
)  
(classifier): Linear(in_features=2208, out_features=1000, bias=True)  
)  
)
```

In [0]:

```
# Freeze parameters so we don't backprop through them  
for param in densenet.parameters():  
    param.requires_grad = False  
  
n_inputs = densenet.classifier.in_features  
n_classes = len(class_names)  
densenet.classifier = nn.Linear(n_inputs, n_classes, bias = True)  
  
# Unfreeze training for classifier layers  
for param in densenet.classifier.parameters():  
    param.requires_grad = True
```

In [0]:

```
if use_cuda:  
    densenet = densenet.cuda()
```

In [0]:

```
print(densenet)
```

```

DenseNet(
  (features): Sequential(
    (conv0): Conv2d(3, 96, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
    (norm0): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu0): ReLU(inplace=True)
    (pool0): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
    (denseblock1): _DenseBlock(
      (denselayer1): _DenseLayer(
        (norm1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(96, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      )
      (denselayer2): _DenseLayer(
        (norm1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(144, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      )
      (denselayer3): _DenseLayer(
        (norm1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(192, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      )
      (denselayer4): _DenseLayer(
        (norm1): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(240, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      )
      (denselayer5): _DenseLayer(
        (norm1): BatchNorm2d(288, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)

```



```

        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(288, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer6): _DenseLayer(
        (norm1): BatchNorm2d(336, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(336, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
)
(transition1): _Transition(
    (norm): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (relu): ReLU(inplace=True)
    (conv): Conv2d(384, 192, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
    (pool): AvgPool2d(kernel_size=2, stride=2, padding=0)
)
(denseblock2): _DenseBlock(
    (denselayer1): _DenseLayer(
        (norm1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(192, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer2): _DenseLayer(
        (norm1): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(240, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer3): _DenseLayer(
        (norm1): BatchNorm2d(288, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(288, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)

```

```

        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer4): _DenseLayer(
        (norm1): BatchNorm2d(336, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(336, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer5): _DenseLayer(
        (norm1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(384, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer6): _DenseLayer(
        (norm1): BatchNorm2d(432, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(432, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer7): _DenseLayer(
        (norm1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(480, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer8): _DenseLayer(
        (norm1): BatchNorm2d(528, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(528, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True

```

```

ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer9): _DenseLayer(
    (norm1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(576, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer10): _DenseLayer(
    (norm1): BatchNorm2d(624, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(624, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer11): _DenseLayer(
    (norm1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(672, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer12): _DenseLayer(
    (norm1): BatchNorm2d(720, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(720, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    )
    (transition2): _Transition(
    (norm): BatchNorm2d(768, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (relu): ReLU(inplace=True)
    (conv): Conv2d(768, 384, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
    (pool): AvgPool2d(kernel_size=2, stride=2, padding=0)

```

```

)
(denseblock3): _DenseBlock(
  (denselayer1): _DenseLayer(
    (norm1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(384, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  )
  (denselayer2): _DenseLayer(
    (norm1): BatchNorm2d(432, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(432, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  )
  (denselayer3): _DenseLayer(
    (norm1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(480, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  )
  (denselayer4): _DenseLayer(
    (norm1): BatchNorm2d(528, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(528, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  )
  (denselayer5): _DenseLayer(
    (norm1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(576, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  )

```

```

)
(denselayer6): _DenseLayer(
  (norm1): BatchNorm2d(624, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(624, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer7): _DenseLayer(
  (norm1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(672, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer8): _DenseLayer(
  (norm1): BatchNorm2d(720, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(720, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer9): _DenseLayer(
  (norm1): BatchNorm2d(768, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(768, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer10): _DenseLayer(
  (norm1): BatchNorm2d(816, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(816, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)

```

```

(denselayer11): _DenseLayer(
  (norm1): BatchNorm2d(864, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(864, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer12): _DenseLayer(
  (norm1): BatchNorm2d(912, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(912, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer13): _DenseLayer(
  (norm1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(960, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer14): _DenseLayer(
  (norm1): BatchNorm2d(1008, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(1008, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer15): _DenseLayer(
  (norm1): BatchNorm2d(1056, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(1056, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer16): _DenseLayer(

```

```

        (norm1): BatchNorm2d(1104, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1104, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer17): _DenseLayer(
        (norm1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer18): _DenseLayer(
        (norm1): BatchNorm2d(1200, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1200, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer19): _DenseLayer(
        (norm1): BatchNorm2d(1248, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1248, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer20): _DenseLayer(
        (norm1): BatchNorm2d(1296, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1296, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer21): _DenseLayer(
        (norm1): BatchNorm2d(1344, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1344, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )

```

```

rue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1344, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer22): _DenseLayer(
    (norm1): BatchNorm2d(1392, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1392, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer23): _DenseLayer(
    (norm1): BatchNorm2d(1440, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1440, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer24): _DenseLayer(
    (norm1): BatchNorm2d(1488, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1488, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer25): _DenseLayer(
    (norm1): BatchNorm2d(1536, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1536, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer26): _DenseLayer(
    (norm1): BatchNorm2d(1584, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)

```



```

        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1584, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer27): _DenseLayer(
        (norm1): BatchNorm2d(1632, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1632, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer28): _DenseLayer(
        (norm1): BatchNorm2d(1680, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1680, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer29): _DenseLayer(
        (norm1): BatchNorm2d(1728, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1728, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer30): _DenseLayer(
        (norm1): BatchNorm2d(1776, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1776, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer31): _DenseLayer(
        (norm1): BatchNorm2d(1824, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)

```

```

        (conv1): Conv2d(1824, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer32): _DenseLayer(
        (norm1): BatchNorm2d(1872, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1872, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer33): _DenseLayer(
        (norm1): BatchNorm2d(1920, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1920, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer34): _DenseLayer(
        (norm1): BatchNorm2d(1968, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1968, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer35): _DenseLayer(
        (norm1): BatchNorm2d(2016, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(2016, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer36): _DenseLayer(
        (norm1): BatchNorm2d(2064, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(2064, 192, kernel_size=(1, 1), stride=(1,

```

```

1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    )
    (transition3): _Transition(
    (norm): BatchNorm2d(2112, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace=True)
    (conv): Conv2d(2112, 1056, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (pool): AvgPool2d(kernel_size=2, stride=2, padding=0)
    )
    (denseblock4): _DenseBlock(
    (denselayer1): _DenseLayer(
    (norm1): BatchNorm2d(1056, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1056, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer2): _DenseLayer(
    (norm1): BatchNorm2d(1104, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1104, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer3): _DenseLayer(
    (norm1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer4): _DenseLayer(
    (norm1): BatchNorm2d(1200, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1200, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)

```

```

        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer5): _DenseLayer(
        (norm1): BatchNorm2d(1248, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1248, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer6): _DenseLayer(
        (norm1): BatchNorm2d(1296, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1296, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer7): _DenseLayer(
        (norm1): BatchNorm2d(1344, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1344, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer8): _DenseLayer(
        (norm1): BatchNorm2d(1392, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1392, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer9): _DenseLayer(
        (norm1): BatchNorm2d(1440, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1440, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)

```

```

        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer10): _DenseLayer(
        (norm1): BatchNorm2d(1488, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1488, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer11): _DenseLayer(
        (norm1): BatchNorm2d(1536, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1536, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer12): _DenseLayer(
        (norm1): BatchNorm2d(1584, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1584, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer13): _DenseLayer(
        (norm1): BatchNorm2d(1632, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1632, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer14): _DenseLayer(
        (norm1): BatchNorm2d(1680, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1680, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),

```

```

padding=(1, 1), bias=False)
    )
    (denselayer15): _DenseLayer(
        (norm1): BatchNorm2d(1728, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1728, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer16): _DenseLayer(
        (norm1): BatchNorm2d(1776, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1776, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer17): _DenseLayer(
        (norm1): BatchNorm2d(1824, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1824, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer18): _DenseLayer(
        (norm1): BatchNorm2d(1872, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1872, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer19): _DenseLayer(
        (norm1): BatchNorm2d(1920, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1920, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )

```

```

)
(denselayer20): _DenseLayer(
  (norm1): BatchNorm2d(1968, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(1968, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer21): _DenseLayer(
  (norm1): BatchNorm2d(2016, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(2016, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer22): _DenseLayer(
  (norm1): BatchNorm2d(2064, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(2064, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer23): _DenseLayer(
  (norm1): BatchNorm2d(2112, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(2112, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer24): _DenseLayer(
  (norm1): BatchNorm2d(2160, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(2160, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)

```

```
)  
(norm5): BatchNorm2d(2208, eps=1e-05, momentum=0.1, affine=True,  
track_running_stats=True)  
)  
(classifier): Linear(in_features=2208, out_features=12, bias=True)  
)
```

In [0]:

```
# criterion and optimzier  
from torch import optim  
criterion_densenet = nn.CrossEntropyLoss()  
optimizer_densenet = optim.Adam(densenet.classifier.parameters(), lr=0.001)
```


In [0]:

```
densenet, train_loss_coll_densenet, valid_loss_coll_densenet = train(1, 25, np.I  
nf, data_loaders, densenet, optimizer_densenet, criterion_densenet, use_cuda,  
"./checkpoint/current_checkpoint_densenet.pt", "./best_model/best_model_densene  
t.pt")
```

```
/usr/local/lib/python3.6/dist-packages/PIL/Image.py:989: UserWarning: Palette images with Transparency expressed in bytes should be converted to RGBA images
```

```
"Palette images with Transparency expressed in bytes should be "
```

```
Epoch: 1      Train Loss: 0.621933      Valid Loss: 0.261463      Train Acc: 84.42      Valid Acc: 93.35
Validation loss decreased (inf --> 0.261463). Saving model ...
Validation loss decreased (0.261463 --> 0.216937). Saving model ...
Epoch: 3      Train Loss: 0.228576      Valid Loss: 0.192575      Train Acc: 93.09      Valid Acc: 94.44
Validation loss decreased (0.216937 --> 0.192575). Saving model ...
Epoch: 4      Train Loss: 0.207432      Valid Loss: 0.175405      Train Acc: 93.74      Valid Acc: 94.74
Validation loss decreased (0.192575 --> 0.175405). Saving model ...
Epoch: 5      Train Loss: 0.188368      Valid Loss: 0.164516      Train Acc: 94.02      Valid Acc: 94.74
Validation loss decreased (0.175405 --> 0.164516). Saving model ...
Epoch: 6      Train Loss: 0.165812      Valid Loss: 0.188890      Train Acc: 94.90      Valid Acc: 93.66
Epoch: 7      Train Loss: 0.177626      Valid Loss: 0.173835      Train Acc: 94.23      Valid Acc: 94.90
Epoch: 8      Train Loss: 0.150281      Valid Loss: 0.177204      Train Acc: 95.08      Valid Acc: 94.59
Epoch: 9      Train Loss: 0.140786      Valid Loss: 0.182341      Train Acc: 95.40      Valid Acc: 94.44
Epoch: 10     Train Loss: 0.135334      Valid Loss: 0.178014      Train Acc: 95.70      Valid Acc: 95.05
Epoch: 11     Train Loss: 0.128897      Valid Loss: 0.193378      Train Acc: 95.67      Valid Acc: 94.28
Epoch: 12     Train Loss: 0.127536      Valid Loss: 0.195300      Train Acc: 95.80      Valid Acc: 94.44
Epoch: 13     Train Loss: 0.125069      Valid Loss: 0.204326      Train Acc: 95.77      Valid Acc: 93.97
Epoch: 14     Train Loss: 0.127642      Valid Loss: 0.199097      Train Acc: 95.76      Valid Acc: 93.97
Epoch: 15     Train Loss: 0.118907      Valid Loss: 0.187449      Train Acc: 96.02      Valid Acc: 94.13
Epoch: 16     Train Loss: 0.122983      Valid Loss: 0.200878      Train Acc: 95.84      Valid Acc: 94.44
Epoch: 17     Train Loss: 0.123348      Valid Loss: 0.194777      Train Acc: 96.06      Valid Acc: 95.05
Epoch: 18     Train Loss: 0.109139      Valid Loss: 0.201957      Train Acc: 96.47      Valid Acc: 94.74
Epoch: 19     Train Loss: 0.116341      Valid Loss: 0.207337      Train Acc: 95.90      Valid Acc: 93.97
Epoch: 20     Train Loss: 0.099330      Valid Loss: 0.212865      Train Acc: 96.56      Valid Acc: 94.59
Epoch: 21     Train Loss: 0.107654      Valid Loss: 0.199744      Train Acc: 96.41      Valid Acc: 94.44
Epoch: 22     Train Loss: 0.101709      Valid Loss: 0.206193      Train Acc: 96.48      Valid Acc: 94.44
Epoch: 23     Train Loss: 0.095124      Valid Loss: 0.214640      Train Acc: 96.61      Valid Acc: 94.74
Epoch: 24     Train Loss: 0.093758      Valid Loss: 0.195339      Train Acc: 96.86      Valid Acc: 95.05
Epoch: 25     Train Loss: 0.090837      Valid Loss: 0.205584      Train Acc: 96.93      Valid Acc: 94.44
```

In [0]:

```
# load state of saved best model
ckp_path = './best_model/best_model_densenet.pt'
densenet, optimizer_densenet, start_epoch, valid_loss_min, train_loss_coll_densenet, valid_loss_coll_densenet = load_ckp(ckp_path, densenet, optimizer_densenet)
```

In [0]:

```
print("model = ", densenet)
print("optimizer = ", optimizer_densenet)
print("start_epoch = ", start_epoch)
print("valid_loss_min = ", valid_loss_min)
print("valid_loss_min = {:.6f}".format(valid_loss_min))
print("train_loss_coll_densenet = ", train_loss_coll_densenet)
print("valid_loss_coll_densenet = ", valid_loss_coll_densenet)
```

```

model = DenseNet(
  (features): Sequential(
    (conv0): Conv2d(3, 96, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
    (norm0): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu0): ReLU(inplace=True)
    (pool0): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
    (denseblock1): _DenseBlock(
      (denselayer1): _DenseLayer(
        (norm1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(96, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      )
      (denselayer2): _DenseLayer(
        (norm1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(144, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      )
      (denselayer3): _DenseLayer(
        (norm1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(192, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      )
      (denselayer4): _DenseLayer(
        (norm1): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(240, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      )
      (denselayer5): _DenseLayer(
        (norm1): BatchNorm2d(288, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)

```

```

        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(288, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer6): _DenseLayer(
        (norm1): BatchNorm2d(336, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(336, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
)
(transition1): _Transition(
    (norm): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (relu): ReLU(inplace=True)
    (conv): Conv2d(384, 192, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
    (pool): AvgPool2d(kernel_size=2, stride=2, padding=0)
)
(denseblock2): _DenseBlock(
    (denselayer1): _DenseLayer(
        (norm1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(192, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer2): _DenseLayer(
        (norm1): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(240, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer3): _DenseLayer(
        (norm1): BatchNorm2d(288, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(288, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)

```

```

        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer4): _DenseLayer(
        (norm1): BatchNorm2d(336, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(336, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer5): _DenseLayer(
        (norm1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(384, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer6): _DenseLayer(
        (norm1): BatchNorm2d(432, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(432, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer7): _DenseLayer(
        (norm1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(480, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer8): _DenseLayer(
        (norm1): BatchNorm2d(528, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(528, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,

```

```

ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer9): _DenseLayer(
    (norm1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(576, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer10): _DenseLayer(
    (norm1): BatchNorm2d(624, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(624, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer11): _DenseLayer(
    (norm1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(672, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer12): _DenseLayer(
    (norm1): BatchNorm2d(720, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(720, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    )
    (transition2): _Transition(
    (norm): BatchNorm2d(768, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (relu): ReLU(inplace=True)
    (conv): Conv2d(768, 384, kernel_size=(1, 1), stride=(1, 1), bi
as=False)
    (pool): AvgPool2d(kernel_size=2, stride=2, padding=0)

```



```

)
(denseblock3): _DenseBlock(
  (denselayer1): _DenseLayer(
    (norm1): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(384, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  )
  (denselayer2): _DenseLayer(
    (norm1): BatchNorm2d(432, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(432, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  )
  (denselayer3): _DenseLayer(
    (norm1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(480, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  )
  (denselayer4): _DenseLayer(
    (norm1): BatchNorm2d(528, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(528, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  )
  (denselayer5): _DenseLayer(
    (norm1): BatchNorm2d(576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(576, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  )

```

```

)
(denselayer6): _DenseLayer(
  (norm1): BatchNorm2d(624, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(624, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer7): _DenseLayer(
  (norm1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(672, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer8): _DenseLayer(
  (norm1): BatchNorm2d(720, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(720, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer9): _DenseLayer(
  (norm1): BatchNorm2d(768, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(768, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer10): _DenseLayer(
  (norm1): BatchNorm2d(816, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(816, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)

```

```

(denselayer11): _DenseLayer(
  (norm1): BatchNorm2d(864, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(864, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer12): _DenseLayer(
  (norm1): BatchNorm2d(912, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(912, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer13): _DenseLayer(
  (norm1): BatchNorm2d(960, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(960, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer14): _DenseLayer(
  (norm1): BatchNorm2d(1008, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(1008, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer15): _DenseLayer(
  (norm1): BatchNorm2d(1056, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): ReLU(inplace=True)
  (conv1): Conv2d(1056, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): ReLU(inplace=True)
  (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
)
(denselayer16): _DenseLayer(

```

```

        (norm1): BatchNorm2d(1104, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1104, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer17): _DenseLayer(
        (norm1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer18): _DenseLayer(
        (norm1): BatchNorm2d(1200, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1200, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer19): _DenseLayer(
        (norm1): BatchNorm2d(1248, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1248, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer20): _DenseLayer(
        (norm1): BatchNorm2d(1296, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1296, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer21): _DenseLayer(
        (norm1): BatchNorm2d(1344, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)

```

```

rue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1344, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer22): _DenseLayer(
    (norm1): BatchNorm2d(1392, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1392, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer23): _DenseLayer(
    (norm1): BatchNorm2d(1440, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1440, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer24): _DenseLayer(
    (norm1): BatchNorm2d(1488, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1488, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer25): _DenseLayer(
    (norm1): BatchNorm2d(1536, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1536, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer26): _DenseLayer(
    (norm1): BatchNorm2d(1584, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)

```

```

        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1584, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer27): _DenseLayer(
        (norm1): BatchNorm2d(1632, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1632, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer28): _DenseLayer(
        (norm1): BatchNorm2d(1680, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1680, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer29): _DenseLayer(
        (norm1): BatchNorm2d(1728, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1728, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer30): _DenseLayer(
        (norm1): BatchNorm2d(1776, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1776, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer31): _DenseLayer(
        (norm1): BatchNorm2d(1824, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)

```

```

        (conv1): Conv2d(1824, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer32): _DenseLayer(
        (norm1): BatchNorm2d(1872, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1872, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer33): _DenseLayer(
        (norm1): BatchNorm2d(1920, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1920, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer34): _DenseLayer(
        (norm1): BatchNorm2d(1968, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1968, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer35): _DenseLayer(
        (norm1): BatchNorm2d(2016, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(2016, 192, kernel_size=(1, 1), stride=(1,
1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=Tr
ue, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer36): _DenseLayer(
        (norm1): BatchNorm2d(2064, eps=1e-05, momentum=0.1, affine=T
rue, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(2064, 192, kernel_size=(1, 1), stride=(1,

```

```

1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    )
    (transition3): _Transition(
    (norm): BatchNorm2d(2112, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace=True)
    (conv): Conv2d(2112, 1056, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (pool): AvgPool2d(kernel_size=2, stride=2, padding=0)
    )
    (denseblock4): _DenseBlock(
    (denselayer1): _DenseLayer(
    (norm1): BatchNorm2d(1056, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1056, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer2): _DenseLayer(
    (norm1): BatchNorm2d(1104, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1104, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer3): _DenseLayer(
    (norm1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu2): ReLU(inplace=True)
    (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer4): _DenseLayer(
    (norm1): BatchNorm2d(1200, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu1): ReLU(inplace=True)
    (conv1): Conv2d(1200, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)

```



```

        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer5): _DenseLayer(
        (norm1): BatchNorm2d(1248, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1248, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer6): _DenseLayer(
        (norm1): BatchNorm2d(1296, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1296, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer7): _DenseLayer(
        (norm1): BatchNorm2d(1344, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1344, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer8): _DenseLayer(
        (norm1): BatchNorm2d(1392, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1392, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer9): _DenseLayer(
        (norm1): BatchNorm2d(1440, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1440, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)

```

```

        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer10): _DenseLayer(
        (norm1): BatchNorm2d(1488, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1488, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer11): _DenseLayer(
        (norm1): BatchNorm2d(1536, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1536, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer12): _DenseLayer(
        (norm1): BatchNorm2d(1584, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1584, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer13): _DenseLayer(
        (norm1): BatchNorm2d(1632, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1632, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer14): _DenseLayer(
        (norm1): BatchNorm2d(1680, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1680, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1),

```

```

padding=(1, 1), bias=False)
    )
    (denselayer15): _DenseLayer(
        (norm1): BatchNorm2d(1728, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1728, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer16): _DenseLayer(
        (norm1): BatchNorm2d(1776, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1776, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer17): _DenseLayer(
        (norm1): BatchNorm2d(1824, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1824, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer18): _DenseLayer(
        (norm1): BatchNorm2d(1872, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1872, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer19): _DenseLayer(
        (norm1): BatchNorm2d(1920, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(1920, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )

```

```

    )
    (denselayer20): _DenseLayer(
      (norm1): BatchNorm2d(1968, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu1): ReLU(inplace=True)
      (conv1): Conv2d(1968, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu2): ReLU(inplace=True)
      (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer21): _DenseLayer(
      (norm1): BatchNorm2d(2016, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu1): ReLU(inplace=True)
      (conv1): Conv2d(2016, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu2): ReLU(inplace=True)
      (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer22): _DenseLayer(
      (norm1): BatchNorm2d(2064, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu1): ReLU(inplace=True)
      (conv1): Conv2d(2064, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu2): ReLU(inplace=True)
      (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer23): _DenseLayer(
      (norm1): BatchNorm2d(2112, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu1): ReLU(inplace=True)
      (conv1): Conv2d(2112, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu2): ReLU(inplace=True)
      (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )
    (denselayer24): _DenseLayer(
      (norm1): BatchNorm2d(2160, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu1): ReLU(inplace=True)
      (conv1): Conv2d(2160, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (norm2): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu2): ReLU(inplace=True)
      (conv2): Conv2d(192, 48, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    )

```

```

    )
    (norm5): BatchNorm2d(2208, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    )
    (classifier): Linear(in_features=2208, out_features=12, bias=True)
    )
optimizer = Adam (
Parameter Group 0
    amsgrad: False
    betas: (0.9, 0.999)
    eps: 1e-08
    lr: 0.001
    weight_decay: 0
)
start_epoch = 6
valid_loss_min = 0.16451590283164952
valid_loss_min = 0.164516
train_loss_coll_densenet = [0.6219334545777139, 0.2871446195725605
4, 0.22857558386858878, 0.20743213217026749, 0.18836752212734606]
valid_loss_coll_densenet = [0.2614627270462708, 0.2169371551763514,
0.19257476943123286, 0.1754050313608816, 0.16451590283164952]

```

Model VGG16

In [31]:

```
vgg16 = models.vgg16(pretrained=True)
```

Downloading: "https://download.pytorch.org/models/vgg16-397923af.pt
h" to /root/.cache/torch/checkpoints/vgg16-397923af.pth
100%|██████████| 528M/528M [00:22<00:00, 24.4MB/s]

In [0]:

```
print(vgg16)
```

```

VGG(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1))
    (3): ReLU(inplace=True)
    (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, c
eil_mode=False)
    (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1))
    (6): ReLU(inplace=True)
    (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding
=(1, 1))
    (8): ReLU(inplace=True)
    (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, c
eil_mode=False)
    (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (11): ReLU(inplace=True)
    (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (13): ReLU(inplace=True)
    (14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (15): ReLU(inplace=True)
    (16): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (17): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (18): ReLU(inplace=True)
    (19): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (20): ReLU(inplace=True)
    (21): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (22): ReLU(inplace=True)
    (23): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (24): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (25): ReLU(inplace=True)
    (26): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (27): ReLU(inplace=True)
    (28): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (29): ReLU(inplace=True)
    (30): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(7, 7))
  (classifier): Sequential(
    (0): Linear(in_features=25088, out_features=4096, bias=True)
    (1): ReLU(inplace=True)
    (2): Dropout(p=0.5, inplace=False)
    (3): Linear(in_features=4096, out_features=4096, bias=True)
    (4): ReLU(inplace=True)
    (5): Dropout(p=0.5, inplace=False)
    (6): Linear(in_features=4096, out_features=1000, bias=True)

```

```
)  
)
```

In [0]:

```
# Freeze parameters so we don't backprop through them  
for param in vgg16.parameters():  
    param.requires_grad = False  
  
n_inputs = vgg16.classifier[6].in_features  
n_classes = len(class_names)  
vgg16.classifier[6] = nn.Linear(n_inputs, n_classes, bias = True)  
  
# Unfreeze training for classifier layers  
for param in vgg16.classifier.parameters():  
    param.requires_grad = True
```


In [16]:

```
print(vgg16)
```

```

VGG(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1))
    (3): ReLU(inplace=True)
    (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, c
eil_mode=False)
    (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1))
    (6): ReLU(inplace=True)
    (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding
=(1, 1))
    (8): ReLU(inplace=True)
    (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, c
eil_mode=False)
    (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (11): ReLU(inplace=True)
    (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (13): ReLU(inplace=True)
    (14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (15): ReLU(inplace=True)
    (16): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (17): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (18): ReLU(inplace=True)
    (19): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (20): ReLU(inplace=True)
    (21): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (22): ReLU(inplace=True)
    (23): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (24): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (25): ReLU(inplace=True)
    (26): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (27): ReLU(inplace=True)
    (28): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (29): ReLU(inplace=True)
    (30): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(7, 7))
  (classifier): Sequential(
    (0): Linear(in_features=25088, out_features=4096, bias=True)
    (1): ReLU(inplace=True)
    (2): Dropout(p=0.5, inplace=False)
    (3): Linear(in_features=4096, out_features=4096, bias=True)
    (4): ReLU(inplace=True)
    (5): Dropout(p=0.5, inplace=False)
    (6): Linear(in_features=4096, out_features=12, bias=True)

```

)
)

In [33]:

```
if use_cuda:
    vgg16 = vgg16.cuda()

print(vgg16)
```

```

VGG(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1))
    (3): ReLU(inplace=True)
    (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, c
eil_mode=False)
    (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1))
    (6): ReLU(inplace=True)
    (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding
=(1, 1))
    (8): ReLU(inplace=True)
    (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, c
eil_mode=False)
    (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (11): ReLU(inplace=True)
    (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (13): ReLU(inplace=True)
    (14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (15): ReLU(inplace=True)
    (16): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (17): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (18): ReLU(inplace=True)
    (19): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (20): ReLU(inplace=True)
    (21): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (22): ReLU(inplace=True)
    (23): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (24): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (25): ReLU(inplace=True)
    (26): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (27): ReLU(inplace=True)
    (28): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
    (29): ReLU(inplace=True)
    (30): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(7, 7))
  (classifier): Sequential(
    (0): Linear(in_features=25088, out_features=4096, bias=True)
    (1): ReLU(inplace=True)
    (2): Dropout(p=0.5, inplace=False)
    (3): Linear(in_features=4096, out_features=4096, bias=True)
    (4): ReLU(inplace=True)
    (5): Dropout(p=0.5, inplace=False)
    (6): Linear(in_features=4096, out_features=12, bias=True)

```

```
)
)
```

In [0]:

```
# criterion and optimzier
criterion_vgg16 = nn.CrossEntropyLoss()
optimizer_vgg16 = optim.Adam(vgg16.classifier.parameters(), lr=0.001)
```

In [0]:

```
#this training got interrupted, so we are going to load our latest saved checkpo
int and continue training
#vgg16, train_loss_coll_vgg16, valid_loss_coll_vgg16 = train(1, 25, np.Inf, data
_loaders, vgg16, optimizer_vgg16, criterion_vgg16, use_cuda, "./checkpoint/curre
nt_checkpoint_vgg16.pt", "./best_model/best_model_vgg16.pt")
```

```
/usr/local/lib/python3.6/dist-packages/PIL/Image.py:989: UserWarnin
g: Palette images with Transparency expressed in bytes should be con
verted to RGBA images
```

```
"Palette images with Transparency expressed in bytes should be "
```

```
Epoch: 1      Train Loss: 1.423385      Valid Loss: 0.796674      Trai
n Acc: 70.36    Valid Acc: 81.45
Validation loss decreased (inf --> 0.796674). Saving model ...
Epoch: 2      Train Loss: 1.472027      Valid Loss: 0.680259      Trai
n Acc: 76.56    Valid Acc: 84.08
Validation loss decreased (0.796674 --> 0.680259). Saving model ...
Epoch: 3      Train Loss: 1.481787      Valid Loss: 0.749719      Trai
n Acc: 79.11    Valid Acc: 84.23
Epoch: 4      Train Loss: 1.287455      Valid Loss: 0.663754      Trai
n Acc: 81.25    Valid Acc: 83.15
Validation loss decreased (0.680259 --> 0.663754). Saving model ...
```

In [0]:

```
# Training above got interrupted so load the best model and continue
# load state of saved best model
ckp_path = './checkpoint/current_checkpoint_vgg16.pt'
vgg16, optimizer_vgg16, start_epoch, valid_loss_min, train_loss_coll_vgg16, vali
d_loss_coll_vgg16 = load_ckp(ckp_path, vgg16, optimizer_vgg16)
```

In [25]:

```
print("model = ", vgg16)
print("optimizer = ", optimizer_vgg16)
print("start_epoch = ", start_epoch)
print("valid_loss_min = ", valid_loss_min)
print("valid_loss_min = {:.6f}".format(valid_loss_min))
print("train_loss_coll_vgg16 = ", train_loss_coll_vgg16)
print("valid_loss_coll_vgg16 = ", valid_loss_coll_vgg16)
```

```

model = VGG(
    (features): Sequential(
      (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1))
      (1): ReLU(inplace=True)
      (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1))
      (3): ReLU(inplace=True)
      (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, c
eil_mode=False)
      (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=
(1, 1))
      (6): ReLU(inplace=True)
      (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding
=(1, 1))
      (8): ReLU(inplace=True)
      (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, c
eil_mode=False)
      (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
      (11): ReLU(inplace=True)
      (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
      (13): ReLU(inplace=True)
      (14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
      (15): ReLU(inplace=True)
      (16): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
      (17): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
      (18): ReLU(inplace=True)
      (19): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
      (20): ReLU(inplace=True)
      (21): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
      (22): ReLU(inplace=True)
      (23): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
      (24): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
      (25): ReLU(inplace=True)
      (26): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
      (27): ReLU(inplace=True)
      (28): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), paddin
g=(1, 1))
      (29): ReLU(inplace=True)
      (30): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    )
    (avgpool): AdaptiveAvgPool2d(output_size=(7, 7))
    (classifier): Sequential(
      (0): Linear(in_features=25088, out_features=4096, bias=True)
      (1): ReLU(inplace=True)
      (2): Dropout(p=0.5, inplace=False)
      (3): Linear(in_features=4096, out_features=4096, bias=True)
      (4): ReLU(inplace=True)
      (5): Dropout(p=0.5, inplace=False)
      (6): Linear(in_features=4096, out_features=12, bias=True)

```



```
)
)
optimizer = Adam (
Parameter Group 0
    amsgrad: False
    betas: (0.9, 0.999)
    eps: 1e-08
    lr: 0.001
    weight_decay: 0
)
start_epoch = 5
valid_loss_min = 0.6637535650784026
valid_loss_min = 0.663754
train_loss_coll_vgg16 = [1.423385033852865, 1.472026505726111, 1.48
17874242392302, 1.2874554431218763]
valid_loss_coll_vgg16 = [0.7966743584209837, 0.6802585518338115, 0.
7497190611071344, 0.6637535650784026]
```

In [29]:

```
# continue training from epoch 5
vgg16, train_loss_coll_vgg16, valid_loss_coll_vgg16 = train(start_epoch, 25, valid_loss_min, data_loaders, vgg16, optimizer_vgg16, criterion_vgg16, use_cuda,
"./checkpoint/current_checkpoint_vgg16.pt", "./best_model/best_model_vgg16.pt")
```

```
/usr/local/lib/python3.6/dist-packages/PIL/Image.py:989: UserWarning: Palette images with Transparency expressed in bytes should be converted to RGBA images
```

"Palette images with Transparency expressed in bytes should be "

Epoch: 5	Train Loss: 1.266488	Valid Loss: 0.643957	Trai
n Acc: 81.86	Valid Acc: 87.79		
Validation loss decreased (0.663754 --> 0.643957). Saving model ...			
Epoch: 6	Train Loss: 1.098177	Valid Loss: 0.739466	Trai
n Acc: 84.55	Valid Acc: 86.55		
Epoch: 7	Train Loss: 0.927129	Valid Loss: 0.694322	Trai
n Acc: 84.41	Valid Acc: 88.10		
Epoch: 8	Train Loss: 0.951358	Valid Loss: 0.626349	Trai
n Acc: 84.39	Valid Acc: 87.02		
Validation loss decreased (0.643957 --> 0.626349). Saving model ...			
Epoch: 9	Train Loss: 0.754076	Valid Loss: 0.770942	Trai
n Acc: 86.65	Valid Acc: 86.24		
Epoch: 10	Train Loss: 0.950669	Valid Loss: 0.717581	Trai
n Acc: 85.58	Valid Acc: 86.71		
Epoch: 11	Train Loss: 0.879983	Valid Loss: 0.645832	Trai
n Acc: 86.05	Valid Acc: 86.40		
Epoch: 12	Train Loss: 0.815145	Valid Loss: 0.756141	Trai
n Acc: 86.36	Valid Acc: 87.64		
Epoch: 13	Train Loss: 0.709123	Valid Loss: 0.847951	Trai
n Acc: 87.45	Valid Acc: 87.48		
Epoch: 14	Train Loss: 0.790640	Valid Loss: 0.595824	Trai
n Acc: 86.66	Valid Acc: 88.41		
Validation loss decreased (0.626349 --> 0.595824). Saving model ...			
Epoch: 15	Train Loss: 0.767113	Valid Loss: 0.652289	Trai
n Acc: 86.63	Valid Acc: 87.64		
Epoch: 16	Train Loss: 0.817106	Valid Loss: 0.534458	Trai
n Acc: 86.69	Valid Acc: 87.48		
Validation loss decreased (0.595824 --> 0.534458). Saving model ...			
Epoch: 17	Train Loss: 0.748672	Valid Loss: 0.484777	Trai
n Acc: 86.87	Valid Acc: 87.94		
Validation loss decreased (0.534458 --> 0.484777). Saving model ...			
Epoch: 18	Train Loss: 0.788026	Valid Loss: 0.711609	Trai
n Acc: 87.26	Valid Acc: 87.17		
Epoch: 19	Train Loss: 0.680553	Valid Loss: 0.564537	Trai
n Acc: 88.32	Valid Acc: 88.87		
Epoch: 20	Train Loss: 0.580187	Valid Loss: 0.774494	Trai
n Acc: 89.24	Valid Acc: 88.41		
Epoch: 21	Train Loss: 0.739245	Valid Loss: 0.580175	Trai
n Acc: 88.34	Valid Acc: 88.10		
Epoch: 22	Train Loss: 0.743228	Valid Loss: 0.534260	Trai
n Acc: 87.61	Valid Acc: 89.49		
Epoch: 23	Train Loss: 0.558273	Valid Loss: 0.670678	Trai
n Acc: 89.15	Valid Acc: 88.10		
Epoch: 24	Train Loss: 0.805957	Valid Loss: 0.503980	Trai
n Acc: 87.94	Valid Acc: 86.86		
Epoch: 25	Train Loss: 0.646812	Valid Loss: 0.633927	Trai
n Acc: 87.40	Valid Acc: 87.33		

In [0]:

```
# load state of saved best model
ckp_path = './best_model/best_model_vgg16.pt'
vgg16, optimizer_vgg16, start_epoch, valid_loss_min, train_loss_coll_vgg16, valid_loss_coll_vgg16 = load_ckp(ckp_path, vgg16, optimizer_vgg16)
```

Inference

In [116]:

```
# Test the performance of the model with the lowest validation loss on the test data
test(data_loaders, model_transfer_mobilev2, criterion_mobilev2h, use_cuda)
```

```
/usr/local/lib/python3.6/dist-packages/PIL/Image.py:989: UserWarning: Palette images with Transparency expressed in bytes should be converted to RGBA images
  "Palette images with Transparency expressed in bytes should be "
```

Test Loss: 0.301537

Test Accuracy: 90% (595/658)

In [0]:

```
# Test the performance of the model with the lowest validation loss on the test data
test(data_loaders, model_shufflenetv2, criterion_shufflenetv2, use_cuda)
```

```
/usr/local/lib/python3.6/dist-packages/PIL/Image.py:989: UserWarning: Palette images with Transparency expressed in bytes should be converted to RGBA images
  "Palette images with Transparency expressed in bytes should be "
```

Test Loss: 1.068708

Test Accuracy: 70% (465/658)

In [137]:

```
# Test the performance of the model with the lowest validation loss on the test data
test(data_loaders, mnasnet, criterion_mnasnet, use_cuda)
```

```
/usr/local/lib/python3.6/dist-packages/PIL/Image.py:989: UserWarning: Palette images with Transparency expressed in bytes should be converted to RGBA images
  "Palette images with Transparency expressed in bytes should be "
```

Test Loss: 0.368574

Test Accuracy: 89% (587/658)

In [0]:

```
# Test the performance of the model with the lowest validation loss on the test data
test(data_loaders, densenet, criterion_densenet, use_cuda)
```

```
/usr/local/lib/python3.6/dist-packages/PIL/Image.py:989: UserWarning: Palette images with Transparency expressed in bytes should be converted to RGBA images
```

```
"Palette images with Transparency expressed in bytes should be "
```

```
Test Loss: 0.196492
```

```
Test Accuracy: 93% (616/658)
```

In [32]:

```
# Test the performance of the model with the lowest validation loss on the test data
test(data_loaders, vgg16, criterion_vgg16, use_cuda)
```

```
/usr/local/lib/python3.6/dist-packages/PIL/Image.py:989: UserWarning: Palette images with Transparency expressed in bytes should be converted to RGBA images
```

```
"Palette images with Transparency expressed in bytes should be "
```

```
Test Loss: 0.941419
```

```
Test Accuracy: 86% (569/658)
```

Summary

In [0]:

```
# load state of saved current checkpoint
ckp_path = './checkpoint/current_checkpoint_mnasnet.pt'
mnasnet, optimizer_mnasnet, start_epoch, valid_loss_min, train_loss_coll_mnasnet, valid_loss_coll_mnasnet = load_ckp(ckp_path, mnasnet, optimizer_mnasnet)
```

In [0]:

```
# load state of saved current checkpoint
ckp_path = './checkpoint/current_checkpoint_densenet.pt'
densenet, optimizer_densenet, start_epoch, valid_loss_min, train_loss_coll_densenet, valid_loss_coll_densenet = load_ckp(ckp_path, densenet, optimizer_densenet)
```

In [0]:

```
# load state of saved current checkpoint
ckp_path = './checkpoint/current_checkpoint_vgg16.pt'
vgg16, optimizer_vgg16, start_epoch, valid_loss_min, train_loss_coll_vgg16, valid_loss_coll_vgg16 = load_ckp(ckp_path, vgg16, optimizer_vgg16)
```

In [37]:

```
print(train_loss_coll_vgg16)
```

```
[1.2664876123877327, 1.0981771974516734, 0.9271289729348886, 0.95135
75905294097, 0.7540757357219289, 0.9506688479299953, 0.8799829916034
887, 0.8151445293086059, 0.709122548881991, 0.7906396783790408, 0.76
71132952045139, 0.8171055247595836, 0.7486723287753834, 0.7880257708
501066, 0.6805530663895599, 0.5801869748128083, 0.739244789671689,
0.7432284551774986, 0.558272696199686, 0.8059571862553606, 0.6468120
345718859]
```

In [0]:

```
/usr/local/lib/python3.6/dist-packages/PIL/Image.py:989: UserWarning: Palette im
ages with Transparency expressed in bytes should be converted to RGBA images
  "Palette images with Transparency expressed in bytes should be "
Epoch: 1          Train Loss: 1.423385      Valid Loss: 0.796674      Train Acc: 70.36
Valid Acc: 81.45
Validation loss decreased (inf --> 0.796674). Saving model ...
Epoch: 2          Train Loss: 1.472027      Valid Loss: 0.680259      Train Acc: 76.56
Valid Acc: 84.08
Validation loss decreased (0.796674 --> 0.680259). Saving model ...
Epoch: 3          Train Loss: 1.481787      Valid Loss: 0.749719      Train Acc: 79.11
Valid Acc: 84.23
Epoch: 4          Train Loss: 1.287455      Valid Loss: 0.663754      Train Acc: 81.25
Valid Acc: 83.15
Validation loss decreased (0.680259 --> 0.663754). Saving model ...
```

In [41]:

```
#manually added the interrupted training loss
train_loss_coll_vgg16.insert(0, 1.423385)
print(train_loss_coll_vgg16)
```

```
[1.423385, 1.472027, 1.481787, 1.287455, 1.2664876123877327, 1.09817
71974516734, 0.9271289729348886, 0.9513575905294097, 0.7540757357219
289, 0.9506688479299953, 0.8799829916034887, 0.8151445293086059, 0.7
09122548881991, 0.7906396783790408, 0.7671132952045139, 0.8171055247
595836, 0.7486723287753834, 0.7880257708501066, 0.6805530663895599,
0.5801869748128083, 0.739244789671689, 0.7432284551774986, 0.5582726
96199686, 0.8059571862553606, 0.6468120345718859]
```

In [51]:

```
#manually added the interrupted validation loss
del valid_loss_coll_vgg16[1]
#valid_loss_coll_vgg16.insert(0, 0.796674)
print(valid_loss_coll_vgg16)
```

```
[0.796674, 0.680259, 0.749719, 0.663754, 0.6439572198221452, 0.73946
60050985065, 0.6943219211961933, 0.6263493596125607, 0.7709419766478
863, 0.7175809115249163, 0.6458318985918021, 0.7561408526838471, 0.8
479511762466828, 0.595823755283398, 0.6522889671475847, 0.5344583016
274701, 0.48477747055832904, 0.7116085233805929, 0.5645366590210975,
0.774494198019943, 0.5801748325688393, 0.534259793125496, 0.67067755
34189835, 0.5039801951814584, 0.6339266635995744]
```

In [53]:

```
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

epochs = range(0,25)

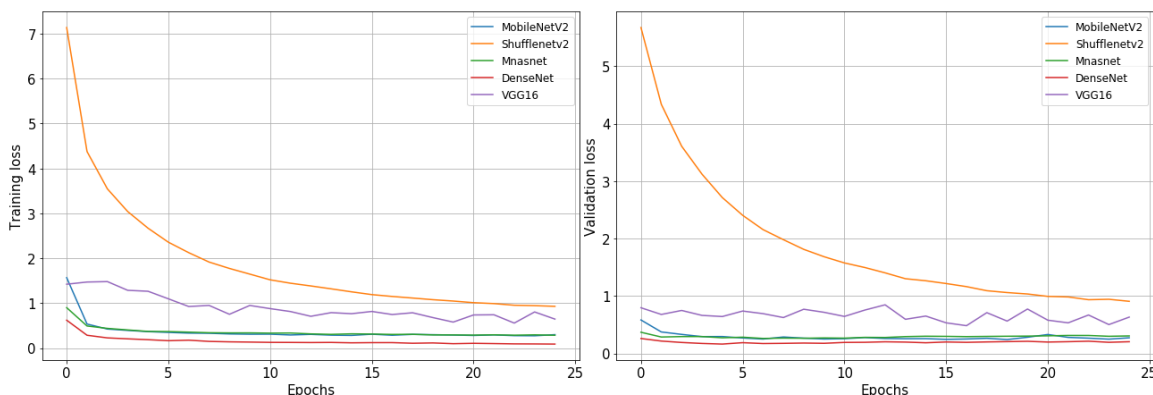
plt.figure()
f, (ax1, ax2) = plt.subplots(1, 2, figsize=(18, 6))
f.tight_layout()
fsize = 15

#ax1.plot(epochs, train_loss_coll_mobilev2, label='MobileNetV2')
ax1.plot(epochs, train_loss_coll_mobilev2_loaded, label='MobileNetV2')
#ax1.plot(epochs, train_loss_coll_shufflenetv2, label='Shufflenetv2')
ax1.plot(epochs, train_loss_coll_shufflenetv2_loaded, label='Shufflenetv2')
ax1.plot(epochs, train_loss_coll_mnasnet, label='Mnasnet')
ax1.plot(epochs, train_loss_coll_densenet, label='DenseNet')
ax1.plot(epochs, train_loss_coll_vgg16, label='VGG16')
ax1.set_xlabel('Epochs', fontsize = fsize)
ax1.set_ylabel('Training loss', fontsize = fsize)
ax1.legend()
ax1.tick_params(labelsize = fsize)
ax1.grid()
ax1.legend(fontsize = 12)

#ax2.plot(epochs, valid_loss_coll_mobilev2, label='MobileNetV2')
ax2.plot(epochs, valid_loss_coll_mobilev2_loaded, label='MobileNetV2')
#ax2.plot(epochs, valid_loss_coll_shufflenetv2, label='Shufflenetv2')
ax2.plot(epochs, valid_loss_coll_shufflenetv2_loaded, label='Shufflenetv2')
ax2.plot(epochs, valid_loss_coll_mnasnet, label='Mnasnet')
ax2.plot(epochs, valid_loss_coll_densenet, label='DenseNet')
ax2.plot(epochs, valid_loss_coll_vgg16, label='VGG16')
ax2.set_xlabel('Epochs', fontsize = fsize)
ax2.set_ylabel('Validation loss', fontsize = fsize)
ax2.legend()
ax2.tick_params(labelsize = fsize)
ax2.grid()
ax2.legend(fontsize = 12)

plt.savefig('Network_architectures.png', bbox_inches='tight', orientation='landscape',
           dpi=300)
```

<Figure size 432x288 with 0 Axes>



Step 4: Write Algorithm

In [0]:

```
from torch.autograd import Variable
def predict_chinese_zodiac_sign_tl_model(img_path, model, use_cuda, class_names):
    # load the image and return the predicted breed
    imsize = 224
    loader = transforms.Compose([transforms.Resize(256),
                                transforms.CenterCrop(imsize),
                                transforms.ToTensor(),
                                transforms.Normalize((0.485, 0.456, 0.406), (0.
229, 0.224, 0.225))])

    image = Image.open(img_path)
    image = loader(image).float()
    image = Variable(image, requires_grad=True)
    image = image.unsqueeze(0)
    if use_cuda:
        image = image.cuda()

    output = model(image)

    _, preds_tensor = torch.max(output, 1) # convert output probabilities to pre
dicted class
    pred = np.squeeze(preds_tensor.numpy()) if not use_cuda else np.squeeze(pred
s_tensor.cpu().numpy())

    return class_names[pred]
```

In [0]:

```
# load state of saved best checkpoint
ckp_path = './best_model/best_model_mnasnet.pt'
best_model, optimizer_best, start_epoch, valid_loss_min, train_loss_coll_best, v
alid_loss_coll_best = load_ckp(ckp_path, mnasnet, optimizer_mnasnet)
```

In [69]:

```
print(valid_loss_min)
```

0.2625579375779426

In [0]:

```
def run_app(img_path):  
  
    title_message = 'Sorry, I don\'t know your chinese zodiac sign!'  
  
    chinese_zodiac_sign = predict_chinese_zodiac_sign_tl_model(img_path, best_model, use_cuda, class_names)  
    title_message = 'Your chinese zodiac sign is ...\\n' + chinese_zodiac_sign  
  
    plt.figure()  
    plt.imshow(Image.open(img_path))  
    plt.title(title_message)  
    plt.show()
```

Step 5: Test Algorithm

In [140]:

```
from glob import glob  
# load filenames for testing  
test_images = np.array(glob("./test_images/*"))  
  
# print number of images in folder  
print('There are %d total test images.' % len(test_images))
```

There are 6 total test images.

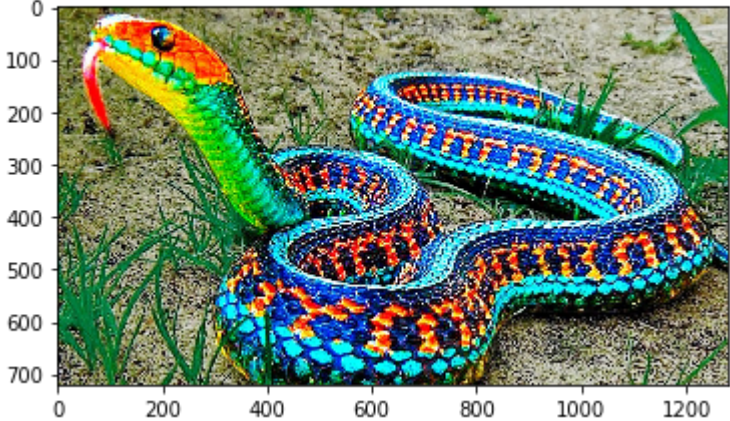
In [141]:

```
n = [0, 1, 2, 3, 4, 5]
for file in np.hstack((test_images[n])):
    run_app(file)
```

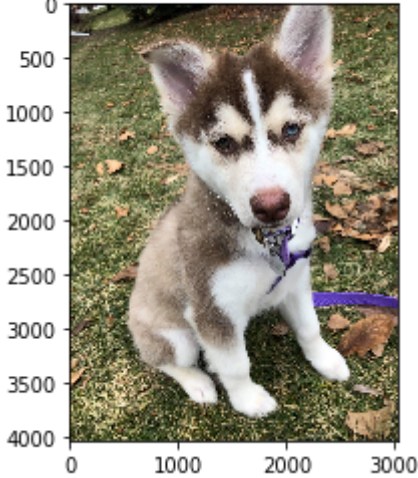
Your chinese zodiac sign is ...
horse



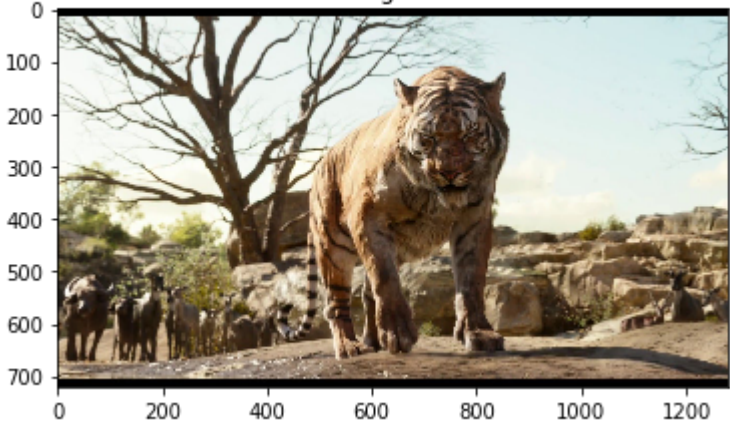
Your chinese zodiac sign is ...
snake



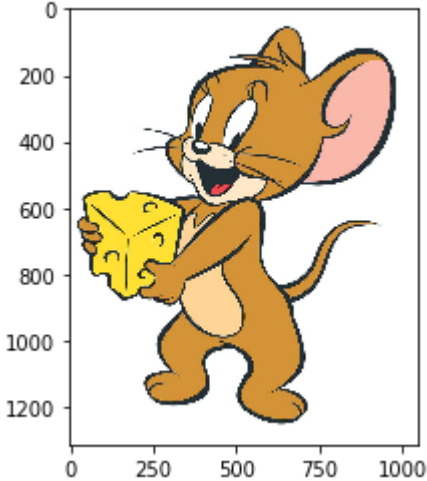
Your chinese zodiac sign is ...
dog



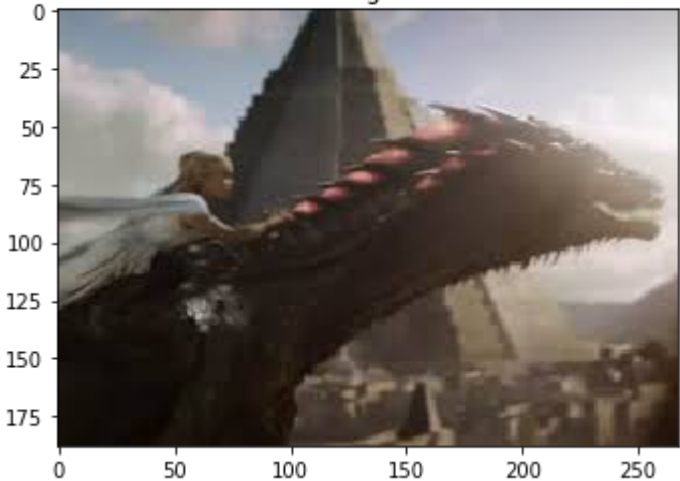
Your chinese zodiac sign is ...
tiger



Your chinese zodiac sign is ...
ratt



Your chinese zodiac sign is ...
dragon



Issue

- RuntimeError: invalid argument 0: Sizes of tensors must match except in dimension 0
 - solution: make width and height of the image is the same; use resize or centerCrop
- Download pytorch pre-train model: URLError: <urlopen error [Errno -3] Temporary failure in name resolution>
 - solution: enable internet connection in kaggle docker
- OSError: cannot identify image file '/kaggle/input/chinese-zodiac-signs/signs/train/rabbit/00000134.jpg'
 - ImageFile.LOAD_TRUNCATED_IMAGES = True