

✓ Hands-on Activity 3.3 - CNN Network Architectures

Technological Institute of the Philippines		Quezon City - Computer Engineering	
Course Code:		CPE 313	
Code Title:		Advanced Machine Learning and Deep Learning	
2nd Semester		AY 2023-2024	
<u>ACTIVITY NO.</u>		Hands-on Activity 3.3 CNN Network Architectures	
Name		Mendoza, Paulo	
Section		CPE32S8	
Date Performed:		March 5, 2024	
Date Submitted:		March 5, 2024	
Instructor:		Engr. Roman M. Richard	

Objective(s):

This activity aims to introduce how to use CNN network architectures

Intended Learning Outcomes (ILOs):

- Demonstrate how to use pretrained model in predicting the object
- Evaluate the accuracy of each pretrained model

Resources:

- Jupyter Notebook
- MNIST

✓ Procedures

Load the necessary libraries

```
# install opencv if you haven't
# pip install opencv-python
# pip install pillow
!pip install scipy
```

```
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (1.11.4)
Requirement already satisfied: numpy<1.28.0,>=1.21.6 in /usr/local/lib/python3.10/dist-p
```

```
from PIL import Image
from keras.preprocessing import image
import numpy as np
import pandas as pd
```

```
import cv2
from scipy.misc.pilutil import imread
```

```
import matplotlib.pyplot as plt
```

```
-----
ModuleNotFoundError                                Traceback (most recent call last)
<ipython-input-8-52f8e61daed4> in <cell line: 7>()
```

```
5
6 import cv2
----> 7 from scipy.misc.pilutil import imread
8
9 import matplotlib.pyplot as plt
```

```
ModuleNotFoundError: No module named 'scipy.misc.pilutil'
```

```
-----
NOTE: If your import is failing due to a missing package, you can
manually install dependencies using either !pip or !apt.
```

```
To view examples of installing some common dependencies, click the
"Open Examples" button below.
```

OPEN EXAMPLES

Create a function that will capture and save the image from your webcam

```
def get_image(camera):
    retval, im = camera.read()
    return im
```

```
def save_webcam_image(img_path):

    try:
        camera_port = 0
        ramp_frames = 10

        camera = cv2.VideoCapture(camera_port)

        for i in range(ramp_frames):
            retval, im_camera = camera.read()

        retval, im_camera = camera.read()

        im = cv2.resize(im_camera, (224, 224)).astype(np.float32)
        cv2.imwrite(img_path, im)
        del (camera)
        return True
    except ValueError as e:
        print("Image Capture Failed")
    return False
```

- Test your function.
- Place a clock or any object
- Display the saved image

```
img_path = "webcam_test_img.png"
```

```
if save_webcam_image(img_path) is False:
    # Webcam not active, use the Dog Image
    img_path = "rocking_chair.jpg"
    print("Using the Test Rocking Chair Image: {}".format(img_path))
```

```
plt.imshow(imread(img_path))
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-12-f41215454500> in <cell line: 1>()
----> 1 plt.imshow(imread(img_path))

NameError: name 'plt' is not defined
```

Use VGG-16 pre-trained model

```
from keras.applications import vgg16
```

```
vgg16_model = vgg16.VGG16(weights='imagenet')  
vgg16_model.summary()
```

Create a function to load model, image, preprocess input and targets

```
def predict_image(model, img_path, preprocess_input_fn, decode_predictions_fn, target_size=(  
  
    img = image.load_img(img_path, target_size=target_size)  
    x = image.img_to_array(img)  
    x = np.expand_dims(x, axis=0)  
    x = preprocess_input_fn(x)  
  
    preds = model.predict(x)  
    predictions_df = pd.DataFrame(decode_predictions_fn(preds, top=10)[0])  
    predictions_df.columns = ["Predicted Class", "Name", "Probability"]  
    return predictions_df
```

Predict the result

```
predict_image(vgg16_model, img_path, vgg16.preprocess_input, vgg16.decode_predictions)
```

Interpret the result of prediction using VGG16 model

✓ type your answer here

Use Resnet50 pretrained model

```
from keras.applications import resnet50  
  
resnet50_model = resnet50.ResNet50(weights='imagenet')  
  
resnet50_model.summary()
```

Predict the result

```
predict_image(resnet50_model, img_path, resnet50.preprocess_input, resnet50.decode_predictic
```

Interpret the result using RESNET50 pretrained model

✓ type your answer here

✓ Supplementary Activity

- Use at least three pretrained model and predict the result using the saved image.
- Compare their accuracy results.

type your code here

Conclusion

✓ type your answer here