UNIVERSITY OF LINCOLN

CMP9137M Advanced Machine Learning Workshop 4:

Regularization in CNNs and CNN based downstream tasks

https://attendance.lincoln.ac.uk

Access Code:

Laboratory of Vision Engineering

Dr. Lei Zhang

lzhang@Lincoln.ac.uk

Aim: workshop 4

The aims of this workshop are

- To gain practical experience to improve the classification performance using Convolutional Neural Networks (CNNs)-with different Regularization techniques and transfer learning.
- To practice CNN based object detection with different architectures.

Task 1: workshop 4

Flower Image classification using the following CNN structure. Task 1 uses a dataset of about 3,700 photos of flowers. The dataset contains 5 sub-directories, one per class:
data from https://storage.googleapis.com/download.tensorflow.org/example_images/flower_photos.tgz


```
model = Sequential([
  layers.experimental.preprocessing.Rescaling(1./255, input_shape=(img_height, img_width, 3)),
  layers.Conv2D(16, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(32, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(64, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Flatten(),
  layers.Dense(128, activation='relu'),
  layers.Dense(num_classes)
])
```

- Training a simple Convolutional Neural Network (CNN) to observe if the overfitting occurs in the training
- Using the Dropout and Batch normalization to solve the overfitting issue and improve the classification performance

Please refer to the tutorials on the https://www.tensorflow.org/tutorials/images/classification
Feel free to use the offline training with your PC or online training with Google Colab notebooks

Task 2: workshop 4

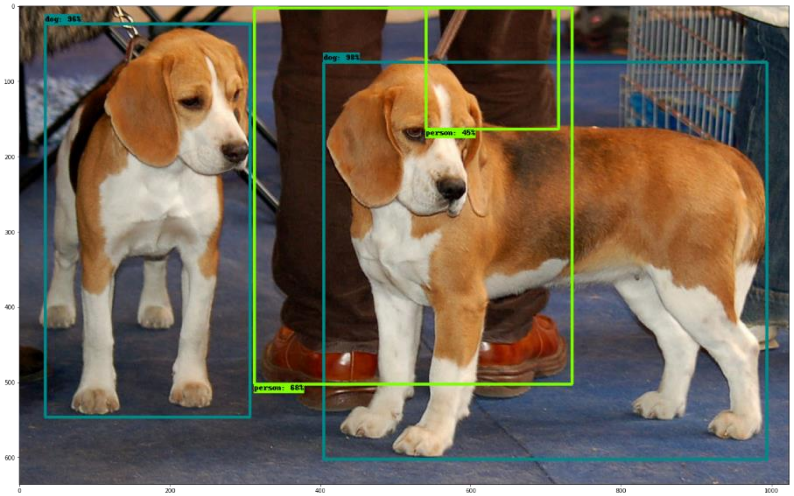
Task 2: Classic Dog&Cat classification. You are required to classify images of cats and dogs by using transfer learning from a pre-trained network. You will use a dataset containing several thousand images of cats and dogs.
data from https://storage.googleapis.com/mledu-datasets/cats_and_dogs_filtered.zip

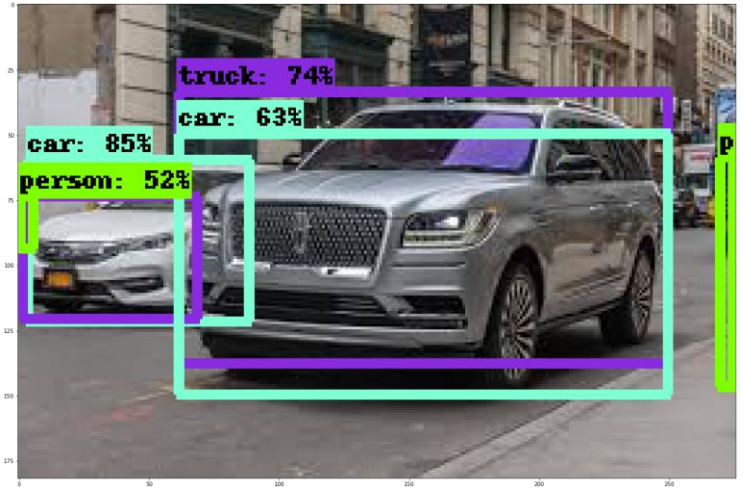
- Create a backbone model (VGG16, ResNet 50, 101, or MobileNetV2) from the pre-trained CNNs with additional classification head (dog&cat).
- Training the model as a feature extractor by freezing the convolutional base. (Train classification head only)
- Fine turning the model (train the convolutional layers in the base model)

Please refer to the tutorials on the https://www.tensorflow.org/tutorials/images/transfer_learning
And API https://www.tensorflow.org/api_docs/python/tf/keras/applications
Feel free to use the offline training with your PC or online training with Google Colab notebooks

Task 3: workshop 4

Object detection using both pre-trained Faster R-CNN model with ResNet 50 backbone and SSD model with ResNet 50. Comparing their detection performances and efficiencies. (Note: You can use your own image to test models)





Please refer to the tutorials on the
https://github.com/tensorflow/docs/blob/master/site/en/hub/tutorials/tf2_object_detection.ipynb
Feel free to use the offline training with your PC or the online training with Google Colab notebooks