Introduction to Artificial Intelligence: Assignment 2 (Version 1.1)

Choose either of the following project:

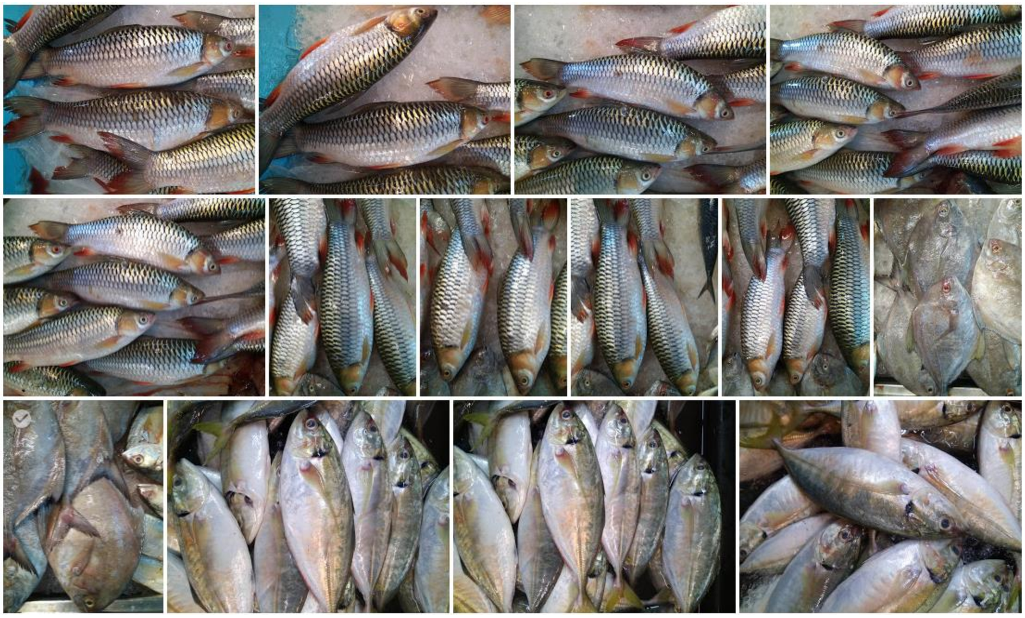
1. **Counting Fish in an image**
2. **Classifying facial attributes for targeted advertisements**

Introduction

This assignment is designed to allow you to put into practise some of the theories learnt in the class lectures especially on machine learning, computer vision and deep learning. This assignment is an opportunity to learn a state-of-the-art deep learning library. This is a group assignment with a maximum of 3.

The core goal of a machine vision system is to achieve a quantifiable, repeatable, and reliable pattern recognition result. The objective of this assignment is to work out a simple system using deep learning neural networks. Deep learning models have been receiving increased attention in recent years. These methods are frequently setting the state-of-the-art (SOTA) in many domains.

1. In this project you will use pictures of fishes taken from the market. These images with act as training and testing data for this project. You are to develop an application to produce the correct count of the fishes in the image. This is useful for fisherman who wants to optimize their time. The fishes may be put on a conveyor belt the system will count the number of fishes. It could also identify the species (this is not required for this task).



2. In this project you will create a classifier that can classify portrait images according to the facial attributes such as having a beard, wearing glasses, cap, long or short hair, etc, and including male or female. One of the uses of this is to create an automatic direct marketing system which will display advertisements based on such facial attributes. For example, if you have long flowing hair, the targeted advertisement may display specialized shampoos. Such systems may be placed in back seat of taxis or hired cars.

Software and Materials

1. **Input data base**
   Some sample images are provided. You should augment with more actual photographs. These may be shared amongst groups. Images from the internet will not be as good. You will need several hundred images to train a good model. You should aim to have your own dataset and do the training required.

2. **Programming language**
   Python only

3. **Deep learning library**
   Please use Tensorflow 2.0 (which includes Keras) You need to install CUDA and cuDNN for the GPU if you have one, then Tensorflow. You can also run the training on the cloud with google colab: https://colab.research.google.com/notebooks/welcome.ipynb . You have about **15 hours** of free use for each google user.

   Ultimately tensorflow can be deployed on Android phones easily.

4. **Hardware platform**
   Windows or Linux, preferably with a good Nvidia GPU with CUDA capability 3.0 or higher. Training on a CPU will be slow could take up to hours.

Requirements (or What you must do)

1. **Images**. Capture, collate and compile your own image database. You will be given some time in the lab to take photographs of the wood samples. You have to decide how to take the photographs, e.g. should they be capture from a consistent height and lighting? How many photographs should I take and what sizes and resolution?

2. **Training and test sets**. Convert the images to input for training and test sets. You have to decide what is the input size and resolution of the images, eg. 256x256 or smaller. Larger will take longer to train and you have less training samples. So you will need to sample them. How will you sample them? You must ensure that there is no overlap between training and test sets. How would you encode the images? Do you need colour or just grayscale images? Do you need to normalize all the images together (Is that reasonable?). How many samples per class? You have to do your own training and not rely on available model weights.

3. **Model**. You are to create your own deep model using Keras.

   Note 1: Many of the earlier computer methods used texture based approach using local binary patterns, co-occurrence matrix, etc. to extract features, we will not use this approach in this project (so don't bother reading about them and don't suggest them).
   Note 2: Also you can't use a pre-trained model.
   Note 3: If you are using a cloud service to train, you need to download the model and weights (h5 format) do that it can run on your local PC. Running test on CPU is fast.

4. **Training**. You need to decide on the training and testing protocol. That is how many epochs to train, what is the learning rate. What weight initialization methods, e.g. Xavier, etc? Do you want to do cross validation? What is the size of the training and test data? You can also extract the convolutional feature maps so that you can understand the model better.

5. **Results and Model assessment.** What result metric should you use? How should the results be reported (eg, confusion table, AUC, precision and recall, F1, etc)? How would you interpret the results? Are the results reasonable? Do you need to modify and re-run the model?

   Note that having a high accuracy model is not a requirement, only that you work on it diligently and reasonably.

6. Conclusion. Summarize and reiterate the key findings and results. Discuss contributions and future works.

7. **Report**. You must **submit** a report that details of points 1-5 above. Please use this format from IEEE: https://www.ieee.org/conferences/publishing/templates.html . Structure it like a conference paper of 5-8 pages. Code must be separated zipped.

Report structure.

1. Introduction (short introduction will do). Explain problem and usefulness of this problem
2. Related work. Review some previous existing work (from academic sources)
3. Approach
   a. Deep learning model description (explain the model structure, requirements 3)
   b. Training and test data (explain how data is collected, sampled, etc, requirements 1& 2)
4. Experiments
   a. Methodology (explain parameters used, how training is conducted, cross validation, training size, epoch, initialization, platform used, etc, requirements 4)
   b. Results and Evaluation (results and explanation, graphs, convolutional feature maps, requirements 5)
5. Conclusion (summary of results, explanation of results, evaluation of the proposed approach, is it practical, etc.)

6. Contributions of each member should be clearly written. This is not counted towards the page total.

Presentation

The group must present the working code and results in class.

Marking Scheme (/100)

1. Appropriate Code and Correctness 15
2. Demo of running code /15
3. Report
    a. Approach /20
    b. Experiments /20
    c. Conclusion /15
4. Presentation /15

Running code must be presented or else zero.

Support

If support needed consult lecturer for advice.