第六章作業講評

主講人：Otto 林弘善

1.推導VoxelNet的輸出

由卷積公式：

$$W' = \left\lfloor \frac{W - F + 2P}{S} + 1 \right\rfloor$$

- W'為卷積後的feature map的大小
- W為卷積前的圖像大小
- F為卷積核的大小
- P是padding的大小
- S是stride的大小

VoxelNet的輸入為：$128 \times 10 \times 400 \times 352$
經過3次CNN分別為：
• Output channel #64, kernel (3, 3, 3), stride (2, 1, 1), padding (1, 1, 1)
• Output channel #64, kernel (3, 3, 3), stride (1, 1, 1), padding (0, 1, 1)
• Output channel #64, kernel (3, 3, 3), stride (2, 1, 1), padding (1, 1, 1)
因此計算每一層的輸出分別為：

- $$\begin{aligned} Layer_1 &= 64 \times \left\lfloor \frac{10 - 3 + 2}{2} + 1 \right\rfloor \times \left\lfloor \frac{400 - 3 + 2}{1} + 1 \right\rfloor \times \left\lfloor \frac{352 - 3 + 2}{1} + 1 \right\rfloor \\ &= 64 \times 5 \times 400 \times 352 \end{aligned}$$
- $$Layer_2 = 64 \times 3 \times 400 \times 352$$
- $$Layer_3 = 64 \times 2 \times 400 \times 352$$

∴ Output is $64 \times 2 \times 400 \times 352$

2.拆分KITTI數據集為train與validation

由課件提供的連結，用 `train.txt` 和 `val.txt` 內容的檔名將KITTI數據分為訓練集和驗證集，代碼實現如下：

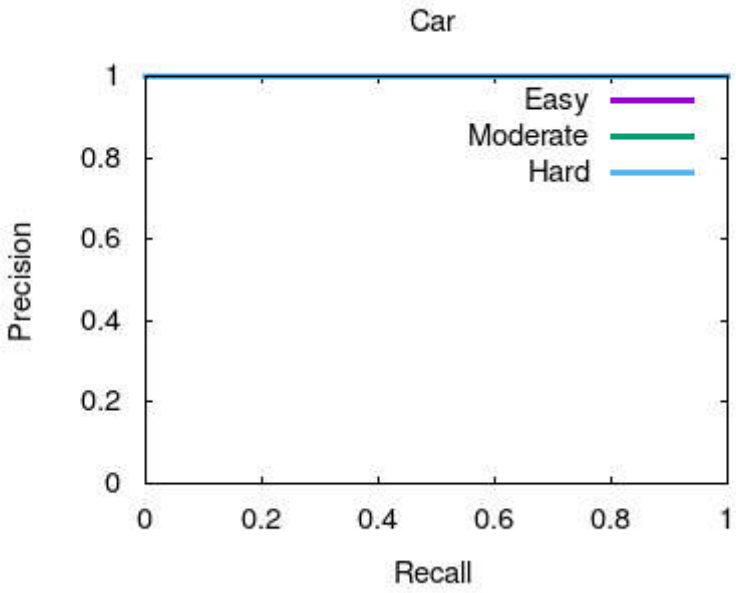
```
def split_data(split_read:str,src:str,dst:str, ftype='train'):
    with open(os.path.join(split_read, ftype+'.txt'), 'r') as ifile:
        for filename in ifile.read().splitlines():
            write_path = os.path.join(dst, ftype)
            if os.path.isdir(write_path) == False:
                os.mkdir(write_path)
            filename += '.txt'
            copyfile(os.path.join(src, filename), os.path.join(write_path, filename))
```

3.對驗證集資料生成目標檢測數據

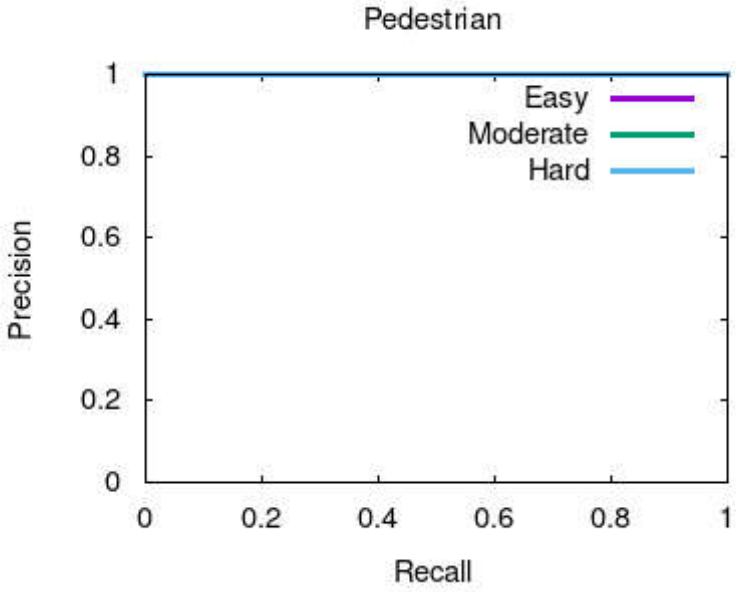
- 仔細閱讀KITTI目標檢測的數據格式，並遵循該文本格式，生成數據；目標檢測數據要對真值每一行加上一個 score 用以生成AP曲線。
  代碼實現：

```
def generate_kitti_obResult(src_dir:str, dst_dir:str, max_score=1.0):
    file_list = get_file_list(src_dir, '.txt')
    if os.path.isdir(dst_dir) == False:
        os.mkdir(dst_dir)
    for file in file_list:
        with open(os.path.join(src_dir, file), 'r') as ifile:
            ob_list = [line.rstrip('\n').split() for line in ifile.readlines()]
            # create kitti detection results
            for i in range(len(ob_list)):
                ob_list[i].append(round(np.random.uniform(0,max_score), 2)) # add score
            with open(os.path.join(dst_dir, file), 'w') as ofile:
                for ob in ob_list:
                    ofile.write(' '.join(map(str, ob))+'\n')
```

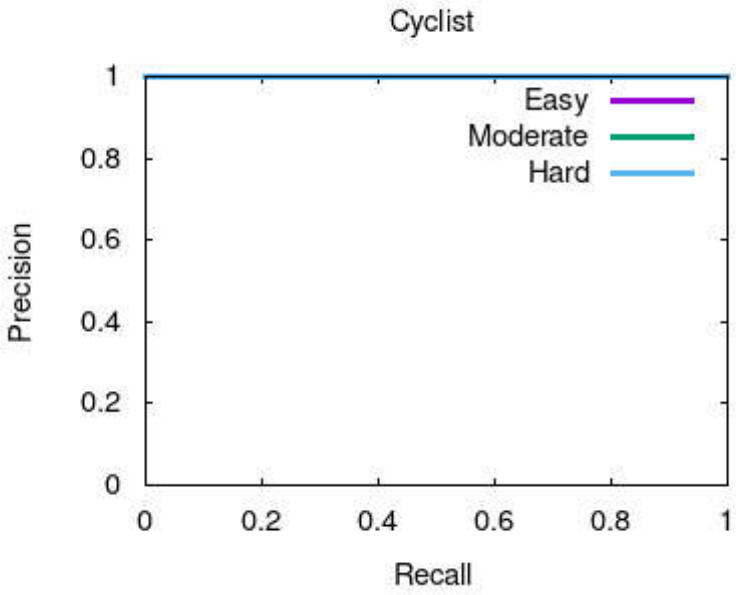
- 繪製AP曲線圖，對真值數據只加上 score 獲得的F1-score為1，由圖可見AP均為最佳值，面積為1，與真值沒有差異。生成的目標檢測資料過於單調，使得AP圖沒有曲線，不呈現Precision與Recall的對比關係，一般來說，Precision和Recall是一個越大另一個就越小的對比關係。以圖中橫座標Recall，縱座標Precision來講，曲線應該會呈現逐漸下滑的趨勢。



圖一 車輛檢測AP



圖二 行人檢測AP



圖三 單車檢測AP