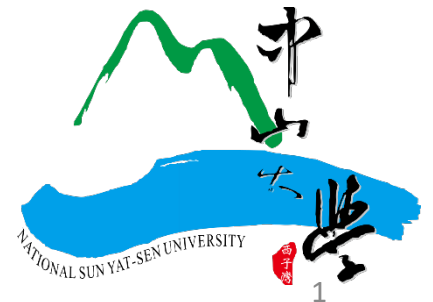


# Assignment 3b

魏家博 (Chia-Po Wei)

Department of Electrical Engineering  
National Sun Yat-sen University





# Outline

1. Kaggle Dogs and Cats Dataset
  - Create Numpy Files from Images
  - Model Fitting with Keras

## Steps for Assignment 3b-1

1. Open [assignment\\_3b.ipynb](#)
2. Create `dogs_cats.pkl` from `train_list.txt` and `test_list.txt`
  1. `train_list.txt` contains 5000 image names for training
  2. `test_list.txt` contains 500 images names for testing
  3. The class label is 0 if the image name has Cat  
The class label is 1 if the image name has Dog
  4. Create `X_train`, `Y_train`, `X_test`, `Y_test`  
`X_train.shape` is of shape `(num_X, 48, 48, 3)`  
`Y_train.shape` is of shape `(num_X,)`  
`X_test.shape` is of shape `(num_Y, 48, 48, 3)`  
`Y_test.shape` is of shape `(num_Y,)`  
`num_X` is less than 5000 because some files are corrupted  
`num_Y` is less than 500 because some files are corrupted
  5. Save `X_train`, `Y_train`, `X_test`, `Y_test` to `dogs_cats.pkl`

### Folder structure

```
+-- current_dir/  
|   +-- PetImages/  
|       +-- Cat/  
|       +-- Dog/
```

### Contents of `train_list.txt`:

```
Cat/10849.jpg  
Dog/9305.jpg  
Cat/10211.jpg  
Dog/12063.jpg  
...
```

# Steps for Assignment 3b-1

See below hints if you are not sure how to complete Step 2.

Implement the following pseudo code (綠色字體是要實作的部分)

```
def get_data_from_file(train_file):  
    train_data = []  
    with open(train_file) as fp:  
        for line in fp:  
            obtain img_path from line  
            try:  
                img = cv2.imread(img_path)  
                img_resized = cv2.resize(img, (48, 48))  
                if line contains Cat  
                    label = 0  
                if line contains Dog  
                    label = 1  
                train_data.append([img_resized, label])  
            except:  
                print error message  
    return train_data
```

## Steps for Assignment 3b-1

- `train_data = get_data_from_file(train_file)`
- `X_train, Y_train = get_image_and_label(train_data)`
- **Implement** `get_image_and_label()` **such that**

`X_train.shape` is of shape `(num_X, 48, 48, 3)`

`Y_train.shape` is of shape `(num_X,)`

`num_X` is the length of `train_data`

- Place below inside `assignment3b_1()`

```
train_data = get_data_from_file(train_file)
X_train, Y_train = get_image_and_label(train_data)
test_data = get_data_from_file(test_file)
X_test, Y_test = get_image_and_label(test_data)
save_path = 'dogs_cats.pkl'
print('Saving to', save_path)
data = {}
data['X_train'] = X_train
data['Y_train'] = Y_train
data['X_test'] = X_test
data['Y_test'] = Y_test
pickle.dump(data, open(save_path, 'wb'))
```

## Steps for Assignment 3b-2

1. Define a Keras Model (need to determine `input_shape` and `output_shape`)

```
def build_model():  
    model = keras.Sequential([  
        keras.layers.Flatten(input_shape=input_shape),  
        keras.layers.Dense(256, activation=tf.nn.relu),  
        keras.layers.Dense(128, activation=tf.nn.relu),  
        keras.layers.Dense(output_shape, activation=tf.nn.softmax)  
    ])  
    return model
```

2. Fit the Keras model to `(X_train, Y_train)` and report the test accuracy.
3. The test accuracy is much lower than the training accuracy. Why?  
How can we improve the test accuracy?

## Steps for Assignment 3b-3

1. Define a Keras Model (need to determine `input_shape` and `output_shape`)

```
def build_model():  
    model = keras.Sequential([  
        keras.layers.Flatten(input_shape=input_shape),  
        keras.layers.Dense(256, activation=tf.nn.relu),  
        keras.layers.Dense(128, activation=tf.nn.relu),  
        keras.layers.Dense(output_shape, activation="linear")  
    ])  
    return model
```

2. Use below to compile your model

```
model.compile(optimizer='adam', loss='mean_squared_error')
```

3. Fit the Keras model to `(X_train, Y_train)`

## Steps for Assignment 3b-3

### 4. Compute and report test accuracy.

Use `reg = model.predict(X_test).squeeze()` to get regression outputs

`reg[i]` is the regression output of `X_test[i]`

- Obtain predictions from `reg`
  - Note that `predictions` and `reg` should have the same shape
  - `predictions[i] = 0` if `reg[i]` is close to 0
  - `predictions[i] = 1` if `reg[i]` is close to 1
- Compute the test accuracy by comparing `predictions` with `Y_test`
  - Use the `==` operator.
  - This task does not require the for loop. You cannot get full marks if you use the for loop.

### 5. Compare the test accuracy of `mean_squared_error` to that of `sparse_categorical_crossentropy`. Which one is better? Why?