Mango contest

Team 10

姚瀚宇 | 王士亞 | 丁憶涵 | 張子宜

Methdology - Deep Learning Classifier

1. Transfer learning (VGG16, VGG19)
2. Our own CNN model

Data:

a. Training data: 5600 images in C1-P1_Train
b. Validation data: 800 images in C1-P1_Dev
c. Testing: upload to the website

Transfer Learning - VGG16, VGG19

tf.keras.applications.VGG16 (19 layers)

- LR = 1e-5 (0.00001)
- Fine tune : layer 5
- Accuracy = 76.06%

tf.keras.applications.VGG19 (22 layers)

- LR = 1e-4 ~ 5e-6 (0.0001 ~ 0.000005)
- Fine tune : layer 8
- Accuracy = 75.37%

Optimizers: Adam, train for 125 epochs

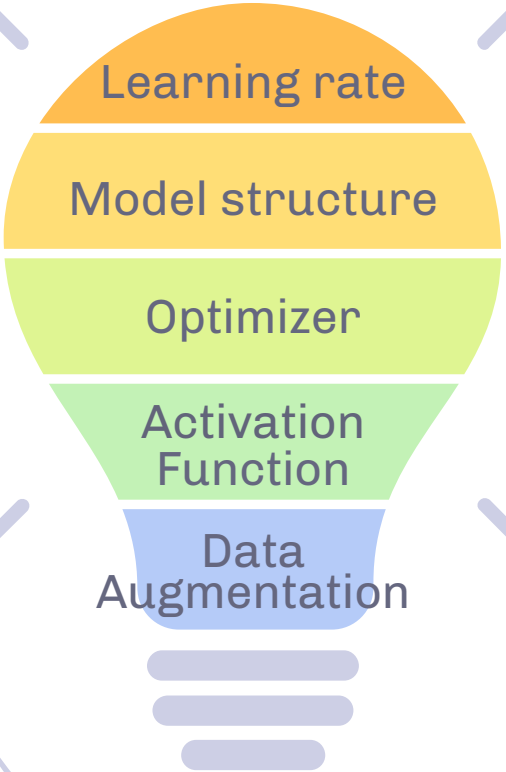
```
model = Sequential()
model.add(Conv2D(32, (3, 3), input_shape=(IMG_SIZE, IMG_SIZE,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(256, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(256, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dropout(0.4))
```

Our own CNN model - Feature Extraction

```
model.add(Dense(512))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.3))

model.add(Dense(512))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.3))

model.add(Dense(NUM_CLASSES, activation='softmax'))

# optimizer = optimizers.Adadelta(lr=LEARNING_RATE, rho=0.95, epsilon=None, decay=0.0001)
optimizer = optimizers.Adam(lr=LEARNING_RATE,beta_1=0.9,beta_2=0.999)
model.compile(loss='categorical_crossentropy', optimizer=optimizer, metrics=['accuracy'])
model.summary()
return model
```

Our own CNN model - Classification

Our own CNN model - Manipulate parameters



Train 5 classifiers

Ensemble Learning - Majority voting

7 models

- model-1: VGG16, acc=76.06%
- model-2: acc=78.93%
- model-3: acc=74.62%
- model-4: VGG19, acc=75.37%
- model-5: acc=78.75%
- model-6: acc=79.43%
- model-7: acc=79.6%

- Model 1 + 2 + 5 + 6 + 7 = 79.56%
- Model 1 + 2 + 6 = 80.12%
- Model 1 + 6 + 7 = 80.56%

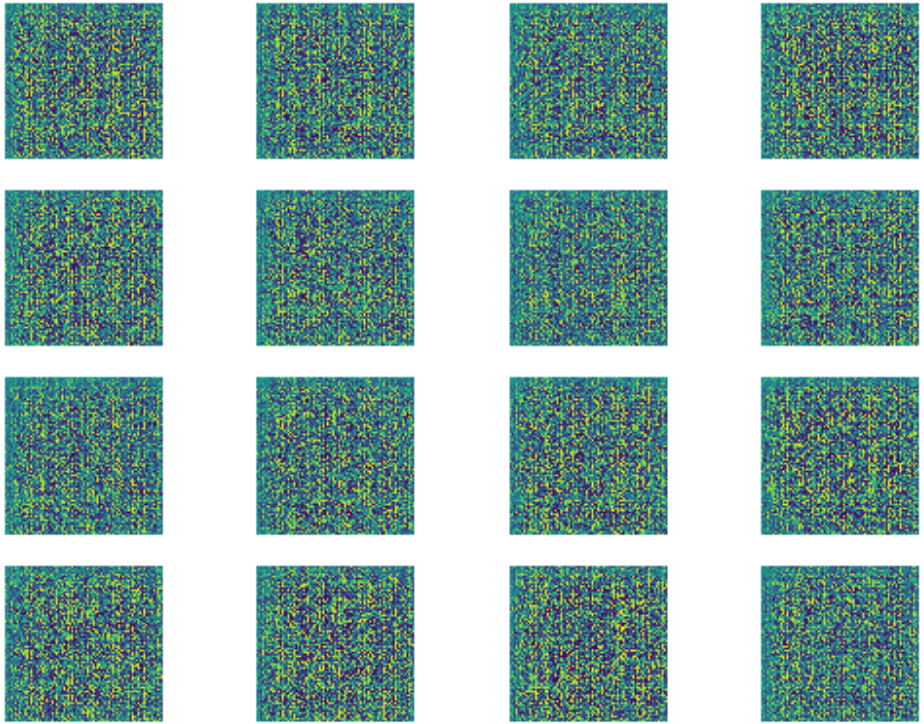
Data Augmentation - ImageDataGenerator

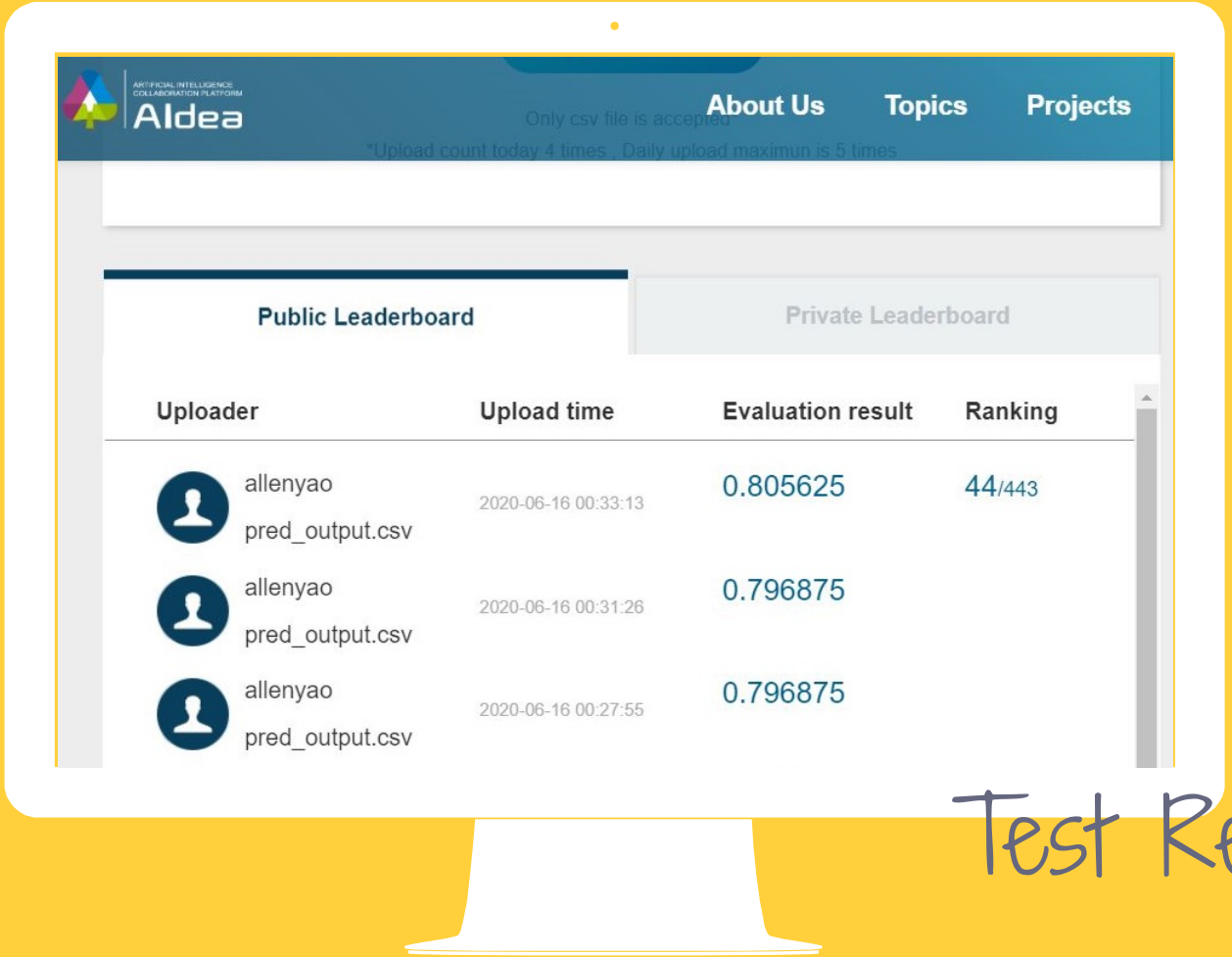
tensorflow.keras.preprocessing.image.ImageDataGenerator

Data Augmentation - ImageDataGenerator

```
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=180,
    width_shift_range=0.2,
    height_shift_range=0.2,
    brightness_range=[0.7, 1.3],
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    vertical_flip=True,
    fill_mode='nearest'
)
```

Data Augmentation - GAN ☹



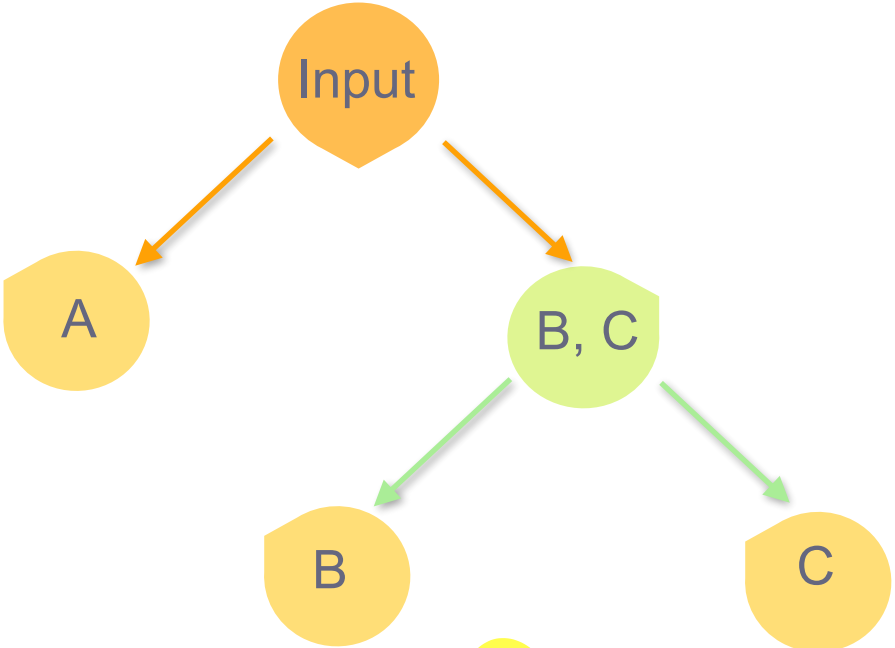

Aldea
About Us
Topics
Projects
Only csv file is accepted
*Upload count today 4 times , Daily upload maximun is 5 times
Public Leaderboard
Private Leaderboard
Uploader
Upload time
Evaluation result
Ranking
allenyao
pred_output.csv
2020-06-16 00:33:13
0.805625
44/443
allenyao
pred_output.csv
2020-06-16 00:31:26
0.796875
allenyao
pred_output.csv
2020-06-16 00:27:55
0.796875

Test Result

Demo Result

```
D:\Class\Junior2\AI\Final\demo_result>python test.py
 image 12 => output: 3 , ans: 2
 image 22 => output: 3 , ans: 2
 image 24 => output: 3 , ans: 2
 image 30 => output: 2 , ans: 1
 image 31 => output: 3 , ans: 2
 image 42 => output: 2 , ans: 1
 image 47 => output: 3 , ans: 2

 53 / 60

 Accuracy: 0.8833333333333333
```

Discussion & Conclusion



Thanks!

Q & A