Experiments on Flower Classification with VGG-16

**Dataset Link:** <https://www.kaggle.com/alxmamaev/flowers-recognition>

**Team Number:** 2

**Team Member:** Furkan Ozbudak

**Notes:**

Abovecommon data augmentation for image pre-processing is applied to all experiments except the raw image experiment.

ImageDataGenerator(

featurewise\_center=False, # set input mean to 0 over the dataset

samplewise\_center=False, # set each sample mean to 0

featurewise\_std\_normalization=False, # divide inputs by std of the dataset

samplewise\_std\_normalization=False, # divide each input by its std

zca\_whitening=False, # apply ZCA whitening

rotation\_range=10, # randomly rotate images in the range (degrees, 0 to 180)

zoom\_range = 0.1, # Randomly zoom image

width\_shift\_range=0.2, # randomly shift images horizontally (fraction of total width)

height\_shift\_range=0.2, # randomly shift images vertically (fraction of total height)

horizontal\_flip=True, # randomly flip images

vertical\_flip=False) # randomly flip images

15 epochs are used for all experiments.

**Experiment 1 (all-frozen)**

**Data Augmentation:**

Images are resized to 150 \* 150.

One-Hot encoding is applied for the class labeling.

Images are divided by 25.

Each image matrix’s pixel values are divided by 255.

25% of the data is taken for validation set. (mistake!)

**Transfer Learning:**

VGG-16:

150 \* 150 input shape and 3 channels.

Pooling is set to average.

Fully connected layers are discarded.

Imagenet weights are used.

All layers are disabled to learn.

**Fine Tuning:**

New fully connected layer with 256 neurons and relu activation.

New fully connected layer with 5 neurons and softmax activation.

**Dropout:**

ReduceLROnPlateau(monitor='val\_acc', factor=0.1, epsilon=0.0001, patience=2, verbose=1)

**Optimizer:**

Adam(lr=1e-4),loss='categorical\_crossentropy',metrics=['accuracy']

**Experiment 2 (all-train)**

**Data Augmentation:**

Images are resized to 150 \* 150.

One-Hot encoding is applied for the class labeling.

Images are divided by 25.

Each image matrix’s pixel values are divided by 255.

25% of the data is taken for validation set. (mistake!)

**Transfer Learning:**

VGG-16:

150 \* 150 input shape and 3 channels.

Pooling is set to average.

Fully connected layers are discarded.

Imagenet weights are used.

All layers are enabled to learn.

**Fine Tuning:**

New fully connected layer with 256 neurons and relu activation.

New fully connected layer with 5 neurons and softmax activation.

**Dropout:**

ReduceLROnPlateau(monitor='val\_acc', factor=0.1, epsilon=0.0001, patience=2, verbose=1)

**Optimizer:**

Adam(lr=1e-4),loss='categorical\_crossentropy',metrics=['accuracy']

**Experiment 3 (last-4-unfrozen)**

**Data Augmentation:**

Images are resized to 150 \* 150.

One-Hot encoding is applied for the class labeling.

Images are divided by 25.

Each image matrix’s pixel values are divided by 255.

5% of the data is taken for validation set.

**Transfer Learning:**

VGG-16:

150 \* 150 input shape and 3 channels.

Pooling is set to average.

Fully connected layers are discarded.

Imagenet weights are used.

Last 4 layers are enabled to learn, rest of the layers are disabled to learn.

**Fine Tuning:**

New fully connected layer with 256 neurons and relu activation.

New fully connected layer with 5 neurons and softmax activation.

**Dropout:**

ReduceLROnPlateau(monitor='val\_acc', factor=0.1, epsilon=0.0001, patience=2, verbose=1)

**Optimizer:**

Adam(lr=1e-4),loss='categorical\_crossentropy',metrics=['accuracy']

**Experiment 4 (raw image)**

**Data Augmentation:**

Images are resized to 150 \* 150.

One-Hot encoding is applied for the class labeling.

Images are divided by 25.

Each image matrix’s pixel values are divided by 255.

25% of the data is taken for validation set.

**Transfer Learning:**

VGG-16:

150 \* 150 input shape and 3 channels.

Pooling is set to average.

Fully connected layers are discarded.

Imagenet weights are used.

Last 4 layers are enabled to learn, rest of the layers are disabled to learn.

**Fine Tuning:**

New fully connected layer with 256 neurons and relu activation.

New fully connected layer with 5 neurons and softmax activation.

**Dropout:**

ReduceLROnPlateau(monitor='val\_acc', factor=0.1, epsilon=0.0001, patience=2, verbose=1)

**Optimizer:**

Adam(lr=1e-4),loss='categorical\_crossentropy',metrics=['accuracy']

**Experiment 5 (image-downscale)**

**Data Augmentation:**

Images are resized to 150 \* 150.

One-Hot encoding is applied for the class labeling.

Images are divided by 25.

Each image matrix’s pixel values are divided by 255.

25% of the data is taken for validation set.

**Transfer Learning:**

VGG-16:

40 \* 40 input shape and 3 channels.

Pooling is set to average.

Fully connected layers are discarded.

Imagenet weights are used.

Last 2 layers are enabled to learn, rest of the layers are disabled to learn.

**Fine Tuning:**

New fully connected layer with 256 neurons and relu activation.

New fully connected layer with 5 neurons and softmax activation.

**Dropout:**

ReduceLROnPlateau(monitor='val\_acc', factor=0.1, epsilon=0.0001, patience=2, verbose=1)

**Optimizer:**

Adam(lr=1e-4),loss='categorical\_crossentropy',metrics=['accuracy']

**Experiment 6 (lower neurons)**

**Data Augmentation:**

Images are resized to 150 \* 150.

One-Hot encoding is applied for the class labeling.

Images are divided by 25.

Each image matrix’s pixel values are divided by 255.

25% of the data is taken for validation set.

**Transfer Learning:**

VGG-16:

40 \* 40 input shape and 3 channels.

Pooling is set to average.

Fully connected layers are discarded.

Imagenet weights are used.

Last 2 layers are enabled to learn, rest of the layers are disabled to learn.

**Fine Tuning:**

New fully connected layer with 50 neurons and relu activation.

New fully connected layer with 5 neurons and softmax activation.

**Dropout:**

ReduceLROnPlateau(monitor='val\_acc', factor=0.1, epsilon=0.0001, patience=2, verbose=1)

**Optimizer:**

Adam(lr=1e-4),loss='categorical\_crossentropy',metrics=['accuracy']

**Results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Experiment Number | Training Loss | Validation Loss | Training Accuracy | Validation Accuracy |
| 1 | 0.81 | 0.84 | 72% | 71% |
| 2 | 0.11 | 0.89 | 96% | 89% |
| 3 | 0.05 | 0.3 | 98% | 90% |
| 4 | 0.02 | 0.6 | 99% | 86% |
| 5 | 0.25 | 0.82 | 91% | 76% |
| 6 | 0.25 | 0.84 | 90% | 76% |