faential_6classes_vggface

June 13, 2023

[1]:
```
!sudo apt-get update
!sudo apt-get install -y gnupg2 curl
!sudo curl -O https://developer.download.nvidia.com/compute/cuda/repos/debian10/
 ↪x86_64/cuda-ubuntu2004.pin
!sudo mv cuda-ubuntu2004.pin /etc/apt/preferences.d/cuda-repository-pin-600
!sudo curl -LO https://developer.download.nvidia.com/compute/cuda/11.4.2/
 ↪local_installers/cuda-repo-debian10-11-4-local_11.4.2-470.57.02-1_amd64.deb
!sudo dpkg -i cuda-repo-debian10-11-4-local_11.4.2-470.57.02-1_amd64.deb
!sudo apt-get update
!sudo apt-get -y install cuda
```

```
Get:1 file:/var/cuda-repo-debian10-11-4-local  InRelease
Ign:1 file:/var/cuda-repo-debian10-11-4-local  InRelease
Get:2 file:/var/cuda-repo-debian10-11-4-local  Release [564 B]
Get:2 file:/var/cuda-repo-debian10-11-4-local  Release [564 B]
Get:3 file:/var/cuda-repo-debian10-11-4-local  Release.gpg [836 B]
Get:3 file:/var/cuda-repo-debian10-11-4-local  Release.gpg [836 B]
Hit:4 http://packages.cloud.google.com/apt gcsfuse-bullseye InRelease
Hit:5 http://security.debian.org/debian-security bullseye-security InRelease
Hit:6 http://deb.debian.org/debian bullseye InRelease
Hit:7 http://packages.cloud.google.com/apt google-compute-engine-bullseye-stable
InRelease
Hit:8 http://deb.debian.org/debian bullseye-updates InRelease
Hit:9 https://packages.cloud.google.com/apt google-fast-socket InRelease
Hit:10 https://download.docker.com/linux/debian bullseye InRelease
Hit:11 http://deb.debian.org/debian bullseye-backports InRelease
Hit:12 http://packages.cloud.google.com/apt cloud-sdk-bullseye InRelease
Hit:13 https://nvidia.github.io/libnvidia-container/stable/debian10/amd64
InRelease
Ign:3 file:/var/cuda-repo-debian10-11-4-local  Release.gpg
Hit:14 https://nvidia.github.io/nvidia-container-runtime/stable/debian10/amd64
InRelease
Hit:15 https://nvidia.github.io/nvidia-docker/debian10/amd64  InRelease
Hit:16 https://packages.cloud.google.com/apt kubernetes-xenial InRelease
Reading package lists… Done
W: GPG error: file:/var/cuda-repo-debian10-11-4-local  Release: The following
signatures couldn't be verified because the public key is not available:
NO_PUBKEY F60F4B3D7FA2AF80
```

```
E: The repository 'file:/var/cuda-repo-debian10-11-4-local  Release' is not
signed.
N: Updating from such a repository can't be done securely, and is therefore
disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration
details.
Reading package lists… Done
E: Unable to parse package file /etc/apt/preferences.d/cuda-repository-pin-600
(1)
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   433  100   433    0     0    436      0 --:--:-- --:--:-- --:--:--   435
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 2432M  100 2432M    0     0   187M      0  0:00:12  0:00:12 --:--:--  199M
(Reading database … 128111 files and directories currently installed.)
Preparing to unpack cuda-repo-debian10-11-4-local_11.4.2-470.57.02-1_amd64.deb
…
Unpacking cuda-repo-debian10-11-4-local (11.4.2-470.57.02-1) over
(11.4.2-470.57.02-1) …
Setting up cuda-repo-debian10-11-4-local (11.4.2-470.57.02-1) …

The public CUDA GPG key does not appear to be installed.
To install the key, run this command:
sudo apt-key add /var/cuda-repo-debian10-11-4-local/7fa2af80.pub

Get:1 file:/var/cuda-repo-debian10-11-4-local  InRelease
Ign:1 file:/var/cuda-repo-debian10-11-4-local  InRelease
Get:2 file:/var/cuda-repo-debian10-11-4-local  Release [564 B]
Get:2 file:/var/cuda-repo-debian10-11-4-local  Release [564 B]
Get:3 file:/var/cuda-repo-debian10-11-4-local  Release.gpg [836 B]
Get:3 file:/var/cuda-repo-debian10-11-4-local  Release.gpg [836 B]
Hit:4 http://deb.debian.org/debian bullseye InRelease
Hit:5 http://packages.cloud.google.com/apt gcsfuse-bullseye InRelease
Hit:6 http://security.debian.org/debian-security bullseye-security InRelease
Hit:7 http://deb.debian.org/debian bullseye-updates InRelease
Hit:8 http://deb.debian.org/debian bullseye-backports InRelease
Hit:9 http://packages.cloud.google.com/apt google-compute-engine-bullseye-stable
InRelease
Hit:10 https://packages.cloud.google.com/apt google-fast-socket InRelease
Hit:11 https://download.docker.com/linux/debian bullseye InRelease
Hit:12 http://packages.cloud.google.com/apt cloud-sdk-bullseye InRelease
Hit:13 https://nvidia.github.io/libnvidia-container/stable/debian10/amd64
InRelease
Ign:3 file:/var/cuda-repo-debian10-11-4-local  Release.gpg
Hit:14 https://nvidia.github.io/nvidia-container-runtime/stable/debian10/amd64
InRelease
Hit:15 https://nvidia.github.io/nvidia-docker/debian10/amd64  InRelease
```

```
Hit:16 https://packages.cloud.google.com/apt kubernetes-xenial InRelease
Reading package lists… Done
W: GPG error: file:/var/cuda-repo-debian10-11-4-local  Release: The following
signatures couldn't be verified because the public key is not available:
NO_PUBKEY F60F4B3D7FA2AF80
E: The repository 'file:/var/cuda-repo-debian10-11-4-local  Release' is not
signed.
N: Updating from such a repository can't be done securely, and is therefore
disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration
details.
Reading package lists… Done
E: Unable to parse package file /etc/apt/preferences.d/cuda-repository-pin-600
(1)
```

[2]:
```
!nvidia-smi
```

```
Tue Jun 13 07:39:29 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 510.47.03    Driver Version: 510.47.03    CUDA Version: 11.6     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   49C    P0    26W /  70W |      0MiB / 15360MiB |     11%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

[3]:
```
!pip install --user tensorflow==2.11.1
```

```
WARNING: Ignoring invalid distribution -eras
(/opt/conda/lib/python3.10/site-packages)
WARNING: Ignoring invalid distribution -rapt
(/opt/conda/lib/python3.10/site-packages)
Requirement already satisfied: tensorflow==2.11.1 in
/home/jupyter/.local/lib/python3.10/site-packages (2.11.1)
Requirement already satisfied: absl-py>=1.0.0 in /opt/conda/lib/python3.10/site-
```

```
packages (from tensorflow==2.11.1) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in
/opt/conda/lib/python3.10/site-packages (from tensorflow==2.11.1) (1.6.3)
Requirement already satisfied: flatbuffers>=2.0 in
/opt/conda/lib/python3.10/site-packages (from tensorflow==2.11.1) (23.3.3)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in
/opt/conda/lib/python3.10/site-packages (from tensorflow==2.11.1) (0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/opt/conda/lib/python3.10/site-packages (from tensorflow==2.11.1) (0.2.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/home/jupyter/.local/lib/python3.10/site-packages (from tensorflow==2.11.1)
(1.54.2)
Requirement already satisfied: h5py>=2.9.0 in /opt/conda/lib/python3.10/site-
packages (from tensorflow==2.11.1) (3.8.0)
Requirement already satisfied: keras<2.12,>=2.11.0 in
/home/jupyter/.local/lib/python3.10/site-packages (from tensorflow==2.11.1)
(2.11.0)
Requirement already satisfied: libclang>=13.0.0 in
/opt/conda/lib/python3.10/site-packages (from tensorflow==2.11.1) (16.0.0)
Requirement already satisfied: numpy>=1.20 in /opt/conda/lib/python3.10/site-
packages (from tensorflow==2.11.1) (1.23.5)
Requirement already satisfied: opt-einsum>=2.3.2 in
/opt/conda/lib/python3.10/site-packages (from tensorflow==2.11.1) (3.3.0)
Requirement already satisfied: packaging in
/home/jupyter/.local/lib/python3.10/site-packages (from tensorflow==2.11.1)
(20.9)
Requirement already satisfied: protobuf<3.20,>=3.9.2 in
/home/jupyter/.local/lib/python3.10/site-packages (from tensorflow==2.11.1)
(3.19.6)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.10/site-
packages (from tensorflow==2.11.1) (67.7.2)
Requirement already satisfied: six>=1.12.0 in /opt/conda/lib/python3.10/site-
packages (from tensorflow==2.11.1) (1.16.0)
Requirement already satisfied: tensorboard<2.12,>=2.11 in
/home/jupyter/.local/lib/python3.10/site-packages (from tensorflow==2.11.1)
(2.11.2)
Requirement already satisfied: tensorflow-estimator<2.12,>=2.11.0 in
/home/jupyter/.local/lib/python3.10/site-packages (from tensorflow==2.11.1)
(2.11.0)
Requirement already satisfied: termcolor>=1.1.0 in
/opt/conda/lib/python3.10/site-packages (from tensorflow==2.11.1) (2.3.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/opt/conda/lib/python3.10/site-packages (from tensorflow==2.11.1) (4.5.0)
Requirement already satisfied: wrapt>=1.11.0 in /opt/conda/lib/python3.10/site-
packages (from tensorflow==2.11.1) (1.14.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/opt/conda/lib/python3.10/site-packages (from tensorflow==2.11.1) (0.29.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
```

```
/opt/conda/lib/python3.10/site-packages (from
astunparse>=1.6.0->tensorflow==2.11.1) (0.40.0)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/opt/conda/lib/python3.10/site-packages (from
tensorboard<2.12,>=2.11->tensorflow==2.11.1) (2.17.3)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
/home/jupyter/.local/lib/python3.10/site-packages (from
tensorboard<2.12,>=2.11->tensorflow==2.11.1) (0.4.6)
Requirement already satisfied: markdown>=2.6.8 in
/opt/conda/lib/python3.10/site-packages (from
tensorboard<2.12,>=2.11->tensorflow==2.11.1) (3.4.3)
Requirement already satisfied: requests<3,>=2.21.0 in
/opt/conda/lib/python3.10/site-packages (from
tensorboard<2.12,>=2.11->tensorflow==2.11.1) (2.28.2)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in
/home/jupyter/.local/lib/python3.10/site-packages (from
tensorboard<2.12,>=2.11->tensorflow==2.11.1) (0.6.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
/opt/conda/lib/python3.10/site-packages (from
tensorboard<2.12,>=2.11->tensorflow==2.11.1) (1.8.1)
Requirement already satisfied: werkzeug>=1.0.1 in
/opt/conda/lib/python3.10/site-packages (from
tensorboard<2.12,>=2.11->tensorflow==2.11.1) (2.1.2)
Requirement already satisfied: pyparsing>=2.0.2 in
/opt/conda/lib/python3.10/site-packages (from packaging->tensorflow==2.11.1)
(3.0.9)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/opt/conda/lib/python3.10/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow==2.11.1) (4.2.4)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/opt/conda/lib/python3.10/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow==2.11.1) (0.2.7)
Requirement already satisfied: rsa<5,>=3.1.4 in /opt/conda/lib/python3.10/site-
packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow==2.11.1) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/opt/conda/lib/python3.10/site-packages (from google-auth-
oauthlib<0.5,>=0.4.1->tensorboard<2.12,>=2.11->tensorflow==2.11.1) (1.3.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/opt/conda/lib/python3.10/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow==2.11.1) (2.1.1)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.10/site-
packages (from requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow==2.11.1)
(3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/opt/conda/lib/python3.10/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow==2.11.1) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in
```

```
/opt/conda/lib/python3.10/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow==2.11.1) (2022.12.7)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in
/opt/conda/lib/python3.10/site-packages (from pyasn1-modules>=0.2.1->google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow==2.11.1) (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in
/opt/conda/lib/python3.10/site-packages (from requests-oauthlib>=0.7.0->google-
auth-oauthlib<0.5,>=0.4.1->tensorboard<2.12,>=2.11->tensorflow==2.11.1) (3.2.2)
WARNING: Ignoring invalid distribution -eras
(/opt/conda/lib/python3.10/site-packages)
WARNING: Ignoring invalid distribution -rapt
(/opt/conda/lib/python3.10/site-packages)
```

[4]:
```
import tensorflow as tf
```

```
2023-06-13 07:39:33.186196: I tensorflow/core/platform/cpu_feature_guard.cc:193]
This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical
operations:  AVX2 AVX512F FMA
To enable them in other operations, rebuild TensorFlow with the appropriate
compiler flags.
2023-06-13 07:39:34.227173: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc:64] Could
not load dynamic library 'libnvinfer.so.7'; dlerror: libnvinfer.so.7: cannot
open shared object file: No such file or directory; LD_LIBRARY_PATH:
/usr/local/cuda/lib64:/usr/local/nccl2/lib:/usr/local/cuda/extras/CUPTI/lib64
2023-06-13 07:39:34.227292: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc:64] Could
not load dynamic library 'libnvinfer_plugin.so.7'; dlerror:
libnvinfer_plugin.so.7: cannot open shared object file: No such file or
directory; LD_LIBRARY_PATH:
/usr/local/cuda/lib64:/usr/local/nccl2/lib:/usr/local/cuda/extras/CUPTI/lib64
2023-06-13 07:39:34.227302: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Cannot
dlopen some TensorRT libraries. If you would like to use Nvidia GPU with
TensorRT, please make sure the missing libraries mentioned above are installed
properly.
```

[5]:
```
print(tf.__version__)
```

```
2.11.1
```

[6]:
```
from keras.layers import Input, Lambda, Dense, Flatten
from keras.models import Model
```

```
from keras.preprocessing import image
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
import numpy as np
from glob import glob
import matplotlib.pyplot as plt
```

[7]:
```
import os
import zipfile
```

define the directories containing your images

[8]:
```
# variabel directory for training pict for each category
train_oily_dir = os.path.join("/home/jupyter/content/faceSkin_tipe_train/oily")
train_normal_dir = os.path.join("/home/jupyter/content/faceSkin_tipe_train/
 ↪normal")
train_combination_dir =os.path.join("/home/jupyter/content/faceSkin_tipe_train/
 ↪combination")
train_sensitive_dir = os.path.join("/home/jupyter/content/faceSkin_tipe_train/
 ↪sensitive")
train_dry_dir = os.path.join("/home/jupyter/content/faceSkin_tipe_train/dry")
train_nonface_dir = os.path.join("/home/jupyter/content/faceSkin_tipe_train/
 ↪nonface/")

# variabel directory for validation pict for each category
validation_oily_dir =os.path.join("/home/jupyter/content/
 ↪faceSkin_tipe_validation/oily")
validation_normal_dir =os.path.join("/home/jupyter/content/
 ↪faceSkin_tipe_validation/normal")
validation_combination_dir =os.path.join("/home/jupyter/content/
 ↪faceSkin_tipe_validation/combination")
validation_sensitive_dir =os.path.join("/home/jupyter/content/
 ↪faceSkin_tipe_validation/sensitive")
validation_dry_dir =os.path.join("/home/jupyter/content/
 ↪faceSkin_tipe_validation/dry")
validation_nonface_dir =os.path.join("/home/jupyter/content/
 ↪faceSkin_tipe_validation/nonface")
```

filenames check in directory

[9]:
```
train_dry_names = os.listdir(train_dry_dir)
train_oily_names = os.listdir(train_oily_dir)
train_normal_names =  os.listdir(train_normal_dir)
train_combination_names =  os.listdir(train_combination_dir)
train_sensitive_names =  os.listdir(train_sensitive_dir)
train_nonface_names =  os.listdir(train_nonface_dir)
```

```
validation_dry_names = os.listdir(validation_dry_dir)
validation_oily_names = os.listdir(validation_oily_dir)
validation_normal_names =  os.listdir(validation_normal_dir)
validation_combination_names =  os.listdir(validation_combination_dir)
validation_sensitive_names =  os.listdir(validation_sensitive_dir)
validation_nonface_names =  os.listdir(validation_nonface_dir)

print(f'TRAIN SET DRY: {train_dry_names[:5]}')
print(f'TRAIN SET OILY: {train_dry_names[:5]}')
print(f'TRAIN SET SENSITIVE: {train_dry_names[:5]}')
print(f'TRAIN SET COMBINATION: {train_dry_names[:5]}')
print(f'TRAIN SET NORMAL: {train_dry_names[:5]}')
print(f'TRAIN SET NONFACE: {train_nonface_names[:5]} \n')

print(f'VALIDATION SET DRY: {validation_dry_names[:5]}')
print(f'VALIDATION SET OILY: {validation_dry_names[:5]}')
print(f'VALIDATION SET SENSITIVE: {validation_dry_names[:5]}')
print(f'VALIDATION SET COMBINATION: {validation_dry_names[:5]}')
print(f'VALIDATION SET NORMAL: {validation_dry_names[:5]}')
print(f'VALIDATION SET NONFACE: {validation_nonface_names[:5]}')
```

```
TRAIN SET DRY: ['f1-006-01.jpg', 'folder(185)3.jpg', 'folder(95)2.jpg',
'folder(224)2.jpg', 'folder(219)2.jpg']
TRAIN SET OILY: ['f1-006-01.jpg', 'folder(185)3.jpg', 'folder(95)2.jpg',
'folder(224)2.jpg', 'folder(219)2.jpg']
TRAIN SET SENSITIVE: ['f1-006-01.jpg', 'folder(185)3.jpg', 'folder(95)2.jpg',
'folder(224)2.jpg', 'folder(219)2.jpg']
TRAIN SET COMBINATION: ['f1-006-01.jpg', 'folder(185)3.jpg', 'folder(95)2.jpg',
'folder(224)2.jpg', 'folder(219)2.jpg']
TRAIN SET NORMAL: ['f1-006-01.jpg', 'folder(185)3.jpg', 'folder(95)2.jpg',
'folder(224)2.jpg', 'folder(219)2.jpg']
TRAIN SET NONFACE: ['000000013201.jpg', '000000017115.jpg', '000000002473.jpg',
'000000024021.jpg', '000000015335.jpg']

VALIDATION SET DRY: ['folder(7)2.jpg', 'dd.jpg', 'folder(72)3.jpg',
'folder(77)3.jpg', '1 (72).jpg']
VALIDATION SET OILY: ['folder(7)2.jpg', 'dd.jpg', 'folder(72)3.jpg',
'folder(77)3.jpg', '1 (72).jpg']
VALIDATION SET SENSITIVE: ['folder(7)2.jpg', 'dd.jpg', 'folder(72)3.jpg',
'folder(77)3.jpg', '1 (72).jpg']
VALIDATION SET COMBINATION: ['folder(7)2.jpg', 'dd.jpg', 'folder(72)3.jpg',
'folder(77)3.jpg', '1 (72).jpg']
VALIDATION SET NORMAL: ['folder(7)2.jpg', 'dd.jpg', 'folder(72)3.jpg',
'folder(77)3.jpg', '1 (72).jpg']
VALIDATION SET NONFACE: ['000000051008.jpg', '000000051976.jpg',
'000000045596.jpg', '000000047010.jpg', '000000045090.jpg']
```

chacking total number of images for each categories in training and validation directories

[10]:
```
print(f'total training oily images: {len(os.listdir(train_oily_dir))}')
print(f'total training dry images: {len(os.listdir(train_dry_dir))}')
print(f'total training normal images: {len(os.listdir(train_normal_dir))}')
print(f'total training combination images: {len(os.
 ↪listdir(train_combination_dir))}')
print(f'total training sensitive images: {len(os.
 ↪listdir(train_sensitive_dir))}')
print(f'total training nonface images: {len(os.listdir(train_nonface_dir))}\n')


print(f'total validation oily images: {len(os.listdir(validation_oily_dir))}')
print(f'total validation dry images: {len(os.listdir(validation_dry_dir))}')
print(f'total validation normal images: {len(os.
 ↪listdir(validation_normal_dir))}')
print(f'total validation combination images: {len(os.
 ↪listdir(validation_combination_dir))}')
print(f'total validation sensitive images: {len(os.
 ↪listdir(validation_sensitive_dir))}')
print(f'total validation nonface images: {len(os.
 ↪listdir(validation_nonface_dir))}')
```

```
total training oily images: 209
total training dry images: 207
total training normal images: 99
total training combination images: 110
total training sensitive images: 202
total training nonface images: 200

total validation oily images: 53
total validation dry images: 50
total validation normal images: 25
total validation combination images: 36
total validation sensitive images: 49
total validation nonface images: 50
```

[ ]:
```

```

1 Data preprocessing

using image data generator

[11]:
```
#punya valiant
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import os

# All images will be rescaled by 1./255
train_datagen = ImageDataGenerator(
```

```
      rescale=1/255,
      rotation_range=40,
      width_shift_range=0.2,
      height_shift_range=0.2,
      shear_range=0.2,
      zoom_range=0.2,
      horizontal_flip=True,
      fill_mode='nearest')

validation_datagen = ImageDataGenerator(rescale=1/255)

# Set the base directory where your data is located
base_directory = '/home/jupyter/content/'

# Get the list of subdirectories (classes) in the training directory
train_classes = [subdir for subdir in os.listdir(os.path.join(base_directory,␣
 ↪'faceSkin_tipe_train')) if os.path.isdir(os.path.join(base_directory,␣
 ↪'faceSkin_tipe_train', subdir))]

# Remove the ".ipynb_checkpoints" class from the list if it exists
if '.ipynb_checkpoints' in train_classes:
    train_classes.remove('.ipynb_checkpoints')

# Flow training images in batches of 128 using train_datagen generator
train_generator = train_datagen.flow_from_directory(
        os.path.join(base_directory, 'faceSkin_tipe_train'),  # This is the␣
 ↪source directory for training images
        target_size=(224, 224),  # All images will be resized to 224x224
        batch_size=32,
        class_mode='categorical',
        classes=train_classes)

# Get the list of subdirectories (classes) in the validation directory
validation_classes = [subdir for subdir in os.listdir(os.path.
 ↪join(base_directory, 'faceSkin_tipe_validation')) if os.path.isdir(os.path.
 ↪join(base_directory, 'faceSkin_tipe_validation', subdir))]

# Remove the ".ipynb_checkpoints" class from the list if it exists
if '.ipynb_checkpoints' in validation_classes:
    validation_classes.remove('.ipynb_checkpoints')

# Flow validation images in batches of 128 using validation_datagen generator
validation_generator = validation_datagen.flow_from_directory(
        os.path.join(base_directory, 'faceSkin_tipe_validation'),  # This is␣
 ↪the source directory for validation images
        target_size=(224, 224),  # All images will be resized to 224x224
        batch_size=16,
```

```
        class_mode='categorical',
        classes=validation_classes)


# Get the class names from the generator's class_indices dictionary
class_names = list(train_generator.class_indices.keys())

# Print the class names
print("Class Names:", class_names)
```

```
Found 1027 images belonging to 6 classes.
Found 263 images belonging to 6 classes.
Class Names: ['oily', 'normal', 'sensitive', 'dry', 'nonface', 'combination']
```

[12]:
```
# Access the class indices
class_indices = train_generator.class_indices

# Print the list of classes
print("List of Classes:")
for class_name, class_index in class_indices.items():
    print(class_name, ":", class_index)
```

```
List of Classes:
oily : 0
normal : 1
sensitive : 2
dry : 3
nonface : 4
combination : 5
```

load pretrained model

[13]:
```
!python --version
```

```
Python 3.10.10
```

[14]:
```
!pip install keras_vggface
```

```
WARNING: Ignoring invalid distribution -eras
(/opt/conda/lib/python3.10/site-packages)
WARNING: Ignoring invalid distribution -rapt
(/opt/conda/lib/python3.10/site-packages)
Requirement already satisfied: keras_vggface in
/opt/conda/lib/python3.10/site-packages (0.6)
Requirement already satisfied: numpy>=1.9.1 in /opt/conda/lib/python3.10/site-
packages (from keras_vggface) (1.23.5)
Requirement already satisfied: scipy>=0.14 in /opt/conda/lib/python3.10/site-
packages (from keras_vggface) (1.9.3)
```

```
Requirement already satisfied: h5py in /opt/conda/lib/python3.10/site-packages
(from keras_vggface) (3.8.0)
Requirement already satisfied: pillow in /opt/conda/lib/python3.10/site-packages
(from keras_vggface) (9.5.0)
Requirement already satisfied: keras in
/home/jupyter/.local/lib/python3.10/site-packages (from keras_vggface) (2.11.0)
Requirement already satisfied: six>=1.9.0 in /opt/conda/lib/python3.10/site-
packages (from keras_vggface) (1.16.0)
Requirement already satisfied: pyyaml in /opt/conda/lib/python3.10/site-packages
(from keras_vggface) (5.4.1)
WARNING: Ignoring invalid distribution -eras
(/opt/conda/lib/python3.10/site-packages)
WARNING: Ignoring invalid distribution -rapt
(/opt/conda/lib/python3.10/site-packages)
```

[15]:
```
!pip install keras_applications
```

```
WARNING: Ignoring invalid distribution -eras
(/opt/conda/lib/python3.10/site-packages)
WARNING: Ignoring invalid distribution -rapt
(/opt/conda/lib/python3.10/site-packages)
Requirement already satisfied: keras_applications in
/opt/conda/lib/python3.10/site-packages (1.0.8)
Requirement already satisfied: numpy>=1.9.1 in /opt/conda/lib/python3.10/site-
packages (from keras_applications) (1.23.5)
Requirement already satisfied: h5py in /opt/conda/lib/python3.10/site-packages
(from keras_applications) (3.8.0)
WARNING: Ignoring invalid distribution -eras
(/opt/conda/lib/python3.10/site-packages)
WARNING: Ignoring invalid distribution -rapt
(/opt/conda/lib/python3.10/site-packages)
```

[16]:
```
!pip show keras
```

```
WARNING: Ignoring invalid distribution -eras
(/opt/conda/lib/python3.10/site-packages)
WARNING: Ignoring invalid distribution -rapt
(/opt/conda/lib/python3.10/site-packages)
Name: keras
```

```
Version: 2.11.0
Summary: Deep learning for humans.
Home-page: https://keras.io/
Author: Keras team
Author-email: keras-users@googlegroups.com
License: Apache 2.0
Location: /home/jupyter/.local/lib/python3.10/site-packages
Requires:
Required-by: keras-vggface, tensorflow
```

[17]:
```
import tensorflow.keras as keras
```

[18]:
```
print(tf.__version__)
```

```
2.11.1
```

[19]:
```
from keras_vggface.vggface import VGGFace
from tensorflow.keras import layers

# Set the weights file you downloaded into a variable
local_weights_file = '/home/jupyter/content/vgg_face_weights.h5'

# Initialize the base model.
# Set the input shape and remove the dense layers.
pre_trained_model = VGGFace(input_shape = (224, 224, 3),
                                include_top = False,
                                model = 'vgg16')

# Load the pre-trained weights you downloaded.
pre_trained_model.load_weights(local_weights_file, by_name=True,␣
 ↪skip_mismatch=True)

# Freeze the weights of the layers.
for layer in pre_trained_model.layers:
  layer.trainable = False
```

[20]:
```
pre_trained_model.summary()
```

```
Model: "vggface_vgg16"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 conv1_1 (Conv2D)            (None, 224, 224, 64)      1792

 conv1_2 (Conv2D)            (None, 224, 224, 64)      36928

 pool1 (MaxPooling2D)        (None, 112, 112, 64)      0
```

```
 conv2_1 (Conv2D)            (None, 112, 112, 128)     73856

 conv2_2 (Conv2D)            (None, 112, 112, 128)     147584

 pool2 (MaxPooling2D)        (None, 56, 56, 128)       0

 conv3_1 (Conv2D)            (None, 56, 56, 256)       295168

 conv3_2 (Conv2D)            (None, 56, 56, 256)       590080

 conv3_3 (Conv2D)            (None, 56, 56, 256)       590080

 pool3 (MaxPooling2D)        (None, 28, 28, 256)       0

 conv4_1 (Conv2D)            (None, 28, 28, 512)       1180160

 conv4_2 (Conv2D)            (None, 28, 28, 512)       2359808

 conv4_3 (Conv2D)            (None, 28, 28, 512)       2359808

 pool4 (MaxPooling2D)        (None, 14, 14, 512)       0

 conv5_1 (Conv2D)            (None, 14, 14, 512)       2359808

 conv5_2 (Conv2D)            (None, 14, 14, 512)       2359808

 conv5_3 (Conv2D)            (None, 14, 14, 512)       2359808

 pool5 (MaxPooling2D)        (None, 7, 7, 512)         0

=================================================================
Total params: 14,714,688
Trainable params: 0
Non-trainable params: 14,714,688
_________________________________________________________________
```

[21]:
```
our_model = tf.keras.models.Sequential(pre_trained_model)
our_model.add(tf.keras.layers.Conv2D(filters = 1024, kernel_size = (3,3),␣
 ↪activation = "relu", padding = "same"))
our_model.add(tf.keras.layers.Conv2D(filters = 512, kernel_size = (3,3),␣
 ↪activation = "relu", padding = "same"))
```

[22]:
```
our_model.summary()
```

```
Model: "sequential"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
```

```
=================================================================
 vggface_vgg16 (Functional)  (None, 7, 7, 512)         14714688

 conv2d (Conv2D)             (None, 7, 7, 1024)        4719616

 conv2d_1 (Conv2D)           (None, 7, 7, 512)         4719104

=================================================================
Total params: 24,153,408
Trainable params: 9,438,720
Non-trainable params: 14,714,688
_________________________________________________________________
```

[24]:
```
# Choose `mixed_7` as the last layer of your base model
last_layer = our_model.get_layer('conv2d')
print('last layer output shape: ', last_layer.output_shape)
last_output = last_layer.output
print('last layer output: ', last_output)
```

```
last layer output shape:  (None, 7, 7, 1024)
last layer output:  KerasTensor(type_spec=TensorSpec(shape=(None, 7, 7, 1024),
dtype=tf.float32, name=None), name='conv2d/Relu:0', description="created by
layer 'conv2d'")
```

2 add dense layer depends on the classification (5 category)

[25]:
```
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras import Model


# Flatten the output layer to 1 dimension
x = layers.Flatten()(last_output)
# Add a fully connected layer with 1,024 hidden units and ReLU activation
x = layers.Dense(1024, activation="ReLU")(x)
# Add a dropout rate of 0.2
x = layers.Dropout(0.2)(x)
# Add a final sigmoid layer for classification
x = layers.Dense(6, activation='softmax')(x)

# Append the dense network to the base model
model = Model(our_model.input, x)

# Print the model summary. See your dense network connected at the end.
model.summary()
```

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
```

```
=================================================================
 vggface_vgg16_input (InputL  [(None, 224, 224, 3)]    0
 ayer)

 vggface_vgg16 (Functional)  (None, 7, 7, 512)         14714688

 conv2d (Conv2D)             (None, 7, 7, 1024)        4719616

 flatten (Flatten)           (None, 50176)             0

 dense (Dense)               (None, 1024)              51381248

 dropout (Dropout)           (None, 1024)              0

 dense_1 (Dense)             (None, 6)                 6150

=================================================================
Total params: 70,821,702
Trainable params: 56,107,014
Non-trainable params: 14,714,688
_________________________________________________________________
```

compiling the model

[26]:
```
from tensorflow.keras.optimizers import Adam
model.compile(loss=tf.keras.losses.CategoricalCrossentropy(),
              optimizer=tf.keras.optimizers.Adam(learning_rate = 0.0001),
              metrics=['accuracy'])

#membuat callback untuk menentukan learning rate terbaik
#lr_scheduler = tf.keras.callbacks.LearningRateScheduler(lambda epoch:1e-4 * 10␣
 ↪**(epoch/20))
```

Training

[27]:
```
len(validation_generator)
```

[27]:
```
17
```

[28]:
```
history = model.fit(
      train_generator,
      steps_per_epoch=len(train_generator),
      epochs=150,
      #callbacks = [lr_scheduler],
      verbose=2,
      validation_data = validation_generator,
      validation_steps=len(validation_generator))
```

```
Epoch 1/150
```

```
33/33 - 81s - loss: 1.4597 - accuracy: 0.4070 - val_loss: 1.2538 - val_accuracy:
0.4791 - 81s/epoch - 2s/step
Epoch 2/150
33/33 - 65s - loss: 1.1511 - accuracy: 0.5209 - val_loss: 1.1242 - val_accuracy:
0.4943 - 65s/epoch - 2s/step
Epoch 3/150
33/33 - 65s - loss: 1.1032 - accuracy: 0.5482 - val_loss: 1.1461 - val_accuracy:
0.5019 - 65s/epoch - 2s/step
Epoch 4/150
33/33 - 64s - loss: 1.0574 - accuracy: 0.5463 - val_loss: 1.1241 - val_accuracy:
0.5323 - 64s/epoch - 2s/step
Epoch 5/150
33/33 - 64s - loss: 1.0179 - accuracy: 0.5618 - val_loss: 1.0852 - val_accuracy:
0.5627 - 64s/epoch - 2s/step
Epoch 6/150
33/33 - 64s - loss: 0.9656 - accuracy: 0.5988 - val_loss: 1.0663 - val_accuracy:
0.5551 - 64s/epoch - 2s/step
Epoch 7/150
33/33 - 64s - loss: 0.9088 - accuracy: 0.6280 - val_loss: 1.0729 - val_accuracy:
0.5779 - 64s/epoch - 2s/step
Epoch 8/150
33/33 - 65s - loss: 0.9173 - accuracy: 0.6144 - val_loss: 1.0203 - val_accuracy:
0.5894 - 65s/epoch - 2s/step
Epoch 9/150
33/33 - 64s - loss: 0.8500 - accuracy: 0.6650 - val_loss: 1.0865 - val_accuracy:
0.5932 - 64s/epoch - 2s/step
Epoch 10/150
33/33 - 65s - loss: 0.9249 - accuracy: 0.6232 - val_loss: 0.9578 - val_accuracy:
0.5856 - 65s/epoch - 2s/step
Epoch 11/150
33/33 - 64s - loss: 0.8237 - accuracy: 0.6670 - val_loss: 1.0333 - val_accuracy:
0.6198 - 64s/epoch - 2s/step
Epoch 12/150
33/33 - 64s - loss: 0.7877 - accuracy: 0.6913 - val_loss: 1.0366 - val_accuracy:
0.6008 - 64s/epoch - 2s/step
Epoch 13/150
33/33 - 64s - loss: 0.7939 - accuracy: 0.6943 - val_loss: 0.9870 - val_accuracy:
0.6046 - 64s/epoch - 2s/step
Epoch 14/150
33/33 - 64s - loss: 0.7415 - accuracy: 0.7225 - val_loss: 1.0818 - val_accuracy:
0.6236 - 64s/epoch - 2s/step
Epoch 15/150
33/33 - 64s - loss: 0.7110 - accuracy: 0.7371 - val_loss: 0.9597 - val_accuracy:
0.5970 - 64s/epoch - 2s/step
Epoch 16/150
33/33 - 64s - loss: 0.6974 - accuracy: 0.7352 - val_loss: 0.9505 - val_accuracy:
0.6046 - 64s/epoch - 2s/step
Epoch 17/150
```

```
33/33 - 64s - loss: 0.6776 - accuracy: 0.7342 - val_loss: 1.2675 - val_accuracy:
0.5703 - 64s/epoch - 2s/step
Epoch 18/150
33/33 - 64s - loss: 0.6460 - accuracy: 0.7400 - val_loss: 0.9887 - val_accuracy:
0.6502 - 64s/epoch - 2s/step
Epoch 19/150
33/33 - 65s - loss: 0.6533 - accuracy: 0.7556 - val_loss: 0.9211 - val_accuracy:
0.6426 - 65s/epoch - 2s/step
Epoch 20/150
33/33 - 64s - loss: 0.6313 - accuracy: 0.7614 - val_loss: 0.9412 - val_accuracy:
0.6312 - 64s/epoch - 2s/step
Epoch 21/150
33/33 - 64s - loss: 0.5919 - accuracy: 0.7712 - val_loss: 0.9678 - val_accuracy:
0.6578 - 64s/epoch - 2s/step
Epoch 22/150
33/33 - 64s - loss: 0.5753 - accuracy: 0.7780 - val_loss: 0.9066 - val_accuracy:
0.6616 - 64s/epoch - 2s/step
Epoch 23/150
33/33 - 64s - loss: 0.5637 - accuracy: 0.7936 - val_loss: 0.9728 - val_accuracy:
0.6388 - 64s/epoch - 2s/step
Epoch 24/150
33/33 - 64s - loss: 0.5439 - accuracy: 0.8101 - val_loss: 0.9285 - val_accuracy:
0.6730 - 64s/epoch - 2s/step
Epoch 25/150
33/33 - 64s - loss: 0.5408 - accuracy: 0.7916 - val_loss: 0.9882 - val_accuracy:
0.6350 - 64s/epoch - 2s/step
Epoch 26/150
33/33 - 64s - loss: 0.4968 - accuracy: 0.8179 - val_loss: 0.9303 - val_accuracy:
0.6730 - 64s/epoch - 2s/step
Epoch 27/150
33/33 - 63s - loss: 0.4973 - accuracy: 0.8257 - val_loss: 0.9369 - val_accuracy:
0.6844 - 63s/epoch - 2s/step
Epoch 28/150
33/33 - 64s - loss: 0.5158 - accuracy: 0.8023 - val_loss: 0.9758 - val_accuracy:
0.6464 - 64s/epoch - 2s/step
Epoch 29/150
33/33 - 64s - loss: 0.4647 - accuracy: 0.8335 - val_loss: 0.8932 - val_accuracy:
0.6844 - 64s/epoch - 2s/step
Epoch 30/150
33/33 - 64s - loss: 0.4388 - accuracy: 0.8315 - val_loss: 0.9864 - val_accuracy:
0.6806 - 64s/epoch - 2s/step
Epoch 31/150
33/33 - 64s - loss: 0.4326 - accuracy: 0.8423 - val_loss: 0.9210 - val_accuracy:
0.6692 - 64s/epoch - 2s/step
Epoch 32/150
33/33 - 64s - loss: 0.3967 - accuracy: 0.8578 - val_loss: 1.0397 - val_accuracy:
0.6654 - 64s/epoch - 2s/step
Epoch 33/150
```

```
33/33 - 63s - loss: 0.3639 - accuracy: 0.8773 - val_loss: 0.9850 - val_accuracy:
0.6692 - 63s/epoch - 2s/step
Epoch 34/150
33/33 - 64s - loss: 0.3861 - accuracy: 0.8685 - val_loss: 1.1108 - val_accuracy:
0.6502 - 64s/epoch - 2s/step
Epoch 35/150
33/33 - 65s - loss: 0.3859 - accuracy: 0.8442 - val_loss: 1.0766 - val_accuracy:
0.6730 - 65s/epoch - 2s/step
Epoch 36/150
33/33 - 64s - loss: 0.3841 - accuracy: 0.8588 - val_loss: 0.9733 - val_accuracy:
0.6730 - 64s/epoch - 2s/step
Epoch 37/150
33/33 - 64s - loss: 0.3447 - accuracy: 0.8676 - val_loss: 0.9092 - val_accuracy:
0.6882 - 64s/epoch - 2s/step
Epoch 38/150
33/33 - 64s - loss: 0.3243 - accuracy: 0.8929 - val_loss: 0.9704 - val_accuracy:
0.6844 - 64s/epoch - 2s/step
Epoch 39/150
33/33 - 64s - loss: 0.3053 - accuracy: 0.8890 - val_loss: 0.9947 - val_accuracy:
0.6920 - 64s/epoch - 2s/step
Epoch 40/150
33/33 - 64s - loss: 0.3098 - accuracy: 0.8948 - val_loss: 0.9600 - val_accuracy:
0.7034 - 64s/epoch - 2s/step
Epoch 41/150
33/33 - 64s - loss: 0.2995 - accuracy: 0.8997 - val_loss: 1.0062 - val_accuracy:
0.6616 - 64s/epoch - 2s/step
Epoch 42/150
33/33 - 65s - loss: 0.3067 - accuracy: 0.8919 - val_loss: 1.0887 - val_accuracy:
0.6768 - 65s/epoch - 2s/step
Epoch 43/150
33/33 - 63s - loss: 0.2986 - accuracy: 0.8958 - val_loss: 1.0407 - val_accuracy:
0.7224 - 63s/epoch - 2s/step
Epoch 44/150
33/33 - 63s - loss: 0.2972 - accuracy: 0.8929 - val_loss: 0.9825 - val_accuracy:
0.6920 - 63s/epoch - 2s/step
Epoch 45/150
33/33 - 64s - loss: 0.2785 - accuracy: 0.9026 - val_loss: 1.0035 - val_accuracy:
0.7110 - 64s/epoch - 2s/step
Epoch 46/150
33/33 - 64s - loss: 0.2591 - accuracy: 0.9143 - val_loss: 1.0179 - val_accuracy:
0.7224 - 64s/epoch - 2s/step
Epoch 47/150
33/33 - 63s - loss: 0.2304 - accuracy: 0.9231 - val_loss: 1.2240 - val_accuracy:
0.6806 - 63s/epoch - 2s/step
Epoch 48/150
33/33 - 63s - loss: 0.2621 - accuracy: 0.9007 - val_loss: 1.0384 - val_accuracy:
0.7110 - 63s/epoch - 2s/step
Epoch 49/150
```

```
33/33 - 63s - loss: 0.2274 - accuracy: 0.9260 - val_loss: 0.9883 - val_accuracy:
0.7072 - 63s/epoch - 2s/step
Epoch 50/150
33/33 - 63s - loss: 0.2141 - accuracy: 0.9328 - val_loss: 1.1862 - val_accuracy:
0.6996 - 63s/epoch - 2s/step
Epoch 51/150
33/33 - 64s - loss: 0.2161 - accuracy: 0.9318 - val_loss: 1.0779 - val_accuracy:
0.7148 - 64s/epoch - 2s/step
Epoch 52/150
33/33 - 65s - loss: 0.2198 - accuracy: 0.9202 - val_loss: 1.0861 - val_accuracy:
0.6996 - 65s/epoch - 2s/step
Epoch 53/150
33/33 - 66s - loss: 0.1967 - accuracy: 0.9406 - val_loss: 1.2105 - val_accuracy:
0.6806 - 66s/epoch - 2s/step
Epoch 54/150
33/33 - 63s - loss: 0.1987 - accuracy: 0.9260 - val_loss: 1.2019 - val_accuracy:
0.7186 - 63s/epoch - 2s/step
Epoch 55/150
33/33 - 63s - loss: 0.2029 - accuracy: 0.9241 - val_loss: 1.1999 - val_accuracy:
0.6882 - 63s/epoch - 2s/step
Epoch 56/150
33/33 - 63s - loss: 0.2097 - accuracy: 0.9289 - val_loss: 1.1759 - val_accuracy:
0.6882 - 63s/epoch - 2s/step
Epoch 57/150
33/33 - 63s - loss: 0.2163 - accuracy: 0.9309 - val_loss: 1.0700 - val_accuracy:
0.7376 - 63s/epoch - 2s/step
Epoch 58/150
33/33 - 63s - loss: 0.2284 - accuracy: 0.9221 - val_loss: 1.3474 - val_accuracy:
0.6730 - 63s/epoch - 2s/step
Epoch 59/150
33/33 - 63s - loss: 0.2007 - accuracy: 0.9309 - val_loss: 1.2759 - val_accuracy:
0.6920 - 63s/epoch - 2s/step
Epoch 60/150
33/33 - 63s - loss: 0.2017 - accuracy: 0.9299 - val_loss: 1.0278 - val_accuracy:
0.7262 - 63s/epoch - 2s/step
Epoch 61/150
33/33 - 64s - loss: 0.1594 - accuracy: 0.9513 - val_loss: 1.1543 - val_accuracy:
0.7148 - 64s/epoch - 2s/step
Epoch 62/150
33/33 - 63s - loss: 0.1746 - accuracy: 0.9455 - val_loss: 1.1713 - val_accuracy:
0.7186 - 63s/epoch - 2s/step
Epoch 63/150
33/33 - 63s - loss: 0.1435 - accuracy: 0.9591 - val_loss: 1.2482 - val_accuracy:
0.7110 - 63s/epoch - 2s/step
Epoch 64/150
33/33 - 63s - loss: 0.1432 - accuracy: 0.9533 - val_loss: 1.1989 - val_accuracy:
0.7300 - 63s/epoch - 2s/step
Epoch 65/150
```

```
33/33 - 63s - loss: 0.1592 - accuracy: 0.9416 - val_loss: 1.1847 - val_accuracy:
0.7110 - 63s/epoch - 2s/step
Epoch 66/150
33/33 - 63s - loss: 0.1498 - accuracy: 0.9523 - val_loss: 1.1832 - val_accuracy:
0.7452 - 63s/epoch - 2s/step
Epoch 67/150
33/33 - 63s - loss: 0.1871 - accuracy: 0.9435 - val_loss: 1.2513 - val_accuracy:
0.6920 - 63s/epoch - 2s/step
Epoch 68/150
33/33 - 63s - loss: 0.1530 - accuracy: 0.9503 - val_loss: 1.2773 - val_accuracy:
0.7072 - 63s/epoch - 2s/step
Epoch 69/150
33/33 - 63s - loss: 0.1380 - accuracy: 0.9620 - val_loss: 1.2051 - val_accuracy:
0.7110 - 63s/epoch - 2s/step
Epoch 70/150
33/33 - 63s - loss: 0.1345 - accuracy: 0.9494 - val_loss: 1.2321 - val_accuracy:
0.7186 - 63s/epoch - 2s/step
Epoch 71/150
33/33 - 64s - loss: 0.1341 - accuracy: 0.9601 - val_loss: 1.3268 - val_accuracy:
0.7034 - 64s/epoch - 2s/step
Epoch 72/150
33/33 - 64s - loss: 0.1799 - accuracy: 0.9445 - val_loss: 1.1474 - val_accuracy:
0.7110 - 64s/epoch - 2s/step
Epoch 73/150
33/33 - 63s - loss: 0.1947 - accuracy: 0.9396 - val_loss: 1.1353 - val_accuracy:
0.7110 - 63s/epoch - 2s/step
Epoch 74/150
33/33 - 63s - loss: 0.1356 - accuracy: 0.9581 - val_loss: 1.2056 - val_accuracy:
0.7338 - 63s/epoch - 2s/step
Epoch 75/150
33/33 - 63s - loss: 0.1755 - accuracy: 0.9357 - val_loss: 1.1835 - val_accuracy:
0.7186 - 63s/epoch - 2s/step
Epoch 76/150
33/33 - 63s - loss: 0.1171 - accuracy: 0.9620 - val_loss: 1.1525 - val_accuracy:
0.7376 - 63s/epoch - 2s/step
Epoch 77/150
33/33 - 63s - loss: 0.1099 - accuracy: 0.9649 - val_loss: 1.2640 - val_accuracy:
0.7338 - 63s/epoch - 2s/step
Epoch 78/150
33/33 - 64s - loss: 0.1257 - accuracy: 0.9581 - val_loss: 1.2071 - val_accuracy:
0.7414 - 64s/epoch - 2s/step
Epoch 79/150
33/33 - 63s - loss: 0.1618 - accuracy: 0.9503 - val_loss: 1.2430 - val_accuracy:
0.7186 - 63s/epoch - 2s/step
Epoch 80/150
33/33 - 63s - loss: 0.2022 - accuracy: 0.9211 - val_loss: 1.2626 - val_accuracy:
0.7072 - 63s/epoch - 2s/step
Epoch 81/150
```

```
33/33 - 63s - loss: 0.1254 - accuracy: 0.9562 - val_loss: 1.3748 - val_accuracy:
0.7224 - 63s/epoch - 2s/step
Epoch 82/150
33/33 - 64s - loss: 0.1384 - accuracy: 0.9503 - val_loss: 1.2462 - val_accuracy:
0.7338 - 64s/epoch - 2s/step
Epoch 83/150
33/33 - 63s - loss: 0.1052 - accuracy: 0.9698 - val_loss: 1.3474 - val_accuracy:
0.7414 - 63s/epoch - 2s/step
Epoch 84/150
33/33 - 63s - loss: 0.1121 - accuracy: 0.9640 - val_loss: 1.3500 - val_accuracy:
0.7224 - 63s/epoch - 2s/step
Epoch 85/150
33/33 - 63s - loss: 0.1222 - accuracy: 0.9572 - val_loss: 1.3622 - val_accuracy:
0.7224 - 63s/epoch - 2s/step
Epoch 86/150
33/33 - 63s - loss: 0.1560 - accuracy: 0.9503 - val_loss: 1.3202 - val_accuracy:
0.7186 - 63s/epoch - 2s/step
Epoch 87/150
33/33 - 63s - loss: 0.1192 - accuracy: 0.9620 - val_loss: 1.3695 - val_accuracy:
0.6958 - 63s/epoch - 2s/step
Epoch 88/150
33/33 - 62s - loss: 0.0896 - accuracy: 0.9718 - val_loss: 1.3514 - val_accuracy:
0.7148 - 62s/epoch - 2s/step
Epoch 89/150
33/33 - 63s - loss: 0.0830 - accuracy: 0.9737 - val_loss: 1.3805 - val_accuracy:
0.7148 - 63s/epoch - 2s/step
Epoch 90/150
33/33 - 63s - loss: 0.2444 - accuracy: 0.9309 - val_loss: 1.1691 - val_accuracy:
0.7376 - 63s/epoch - 2s/step
Epoch 91/150
33/33 - 63s - loss: 0.1140 - accuracy: 0.9649 - val_loss: 1.1260 - val_accuracy:
0.7414 - 63s/epoch - 2s/step
Epoch 92/150
33/33 - 63s - loss: 0.1031 - accuracy: 0.9708 - val_loss: 1.2285 - val_accuracy:
0.7490 - 63s/epoch - 2s/step
Epoch 93/150
33/33 - 63s - loss: 0.0928 - accuracy: 0.9747 - val_loss: 1.3104 - val_accuracy:
0.7338 - 63s/epoch - 2s/step
Epoch 94/150
33/33 - 63s - loss: 0.1029 - accuracy: 0.9630 - val_loss: 1.2610 - val_accuracy:
0.7414 - 63s/epoch - 2s/step
Epoch 95/150
33/33 - 63s - loss: 0.0953 - accuracy: 0.9669 - val_loss: 1.3800 - val_accuracy:
0.7224 - 63s/epoch - 2s/step
Epoch 96/150
33/33 - 64s - loss: 0.1021 - accuracy: 0.9659 - val_loss: 1.2352 - val_accuracy:
0.7376 - 64s/epoch - 2s/step
Epoch 97/150
```

```
33/33 - 64s - loss: 0.0929 - accuracy: 0.9659 - val_loss: 1.3739 - val_accuracy:
0.7376 - 64s/epoch - 2s/step
Epoch 98/150
33/33 - 63s - loss: 0.1324 - accuracy: 0.9533 - val_loss: 1.3296 - val_accuracy:
0.7567 - 63s/epoch - 2s/step
Epoch 99/150
33/33 - 63s - loss: 0.0909 - accuracy: 0.9698 - val_loss: 1.2161 - val_accuracy:
0.7567 - 63s/epoch - 2s/step
Epoch 100/150
33/33 - 64s - loss: 0.1678 - accuracy: 0.9484 - val_loss: 1.3049 - val_accuracy:
0.7376 - 64s/epoch - 2s/step
Epoch 101/150
33/33 - 63s - loss: 0.0882 - accuracy: 0.9766 - val_loss: 1.3236 - val_accuracy:
0.7376 - 63s/epoch - 2s/step
Epoch 102/150
33/33 - 63s - loss: 0.0976 - accuracy: 0.9688 - val_loss: 1.3043 - val_accuracy:
0.7490 - 63s/epoch - 2s/step
Epoch 103/150
33/33 - 63s - loss: 0.1124 - accuracy: 0.9669 - val_loss: 1.4611 - val_accuracy:
0.7148 - 63s/epoch - 2s/step
Epoch 104/150
33/33 - 63s - loss: 0.0941 - accuracy: 0.9708 - val_loss: 1.6989 - val_accuracy:
0.6806 - 63s/epoch - 2s/step
Epoch 105/150
33/33 - 63s - loss: 0.1423 - accuracy: 0.9513 - val_loss: 1.2583 - val_accuracy:
0.7719 - 63s/epoch - 2s/step
Epoch 106/150
33/33 - 63s - loss: 0.1664 - accuracy: 0.9484 - val_loss: 1.2530 - val_accuracy:
0.7757 - 63s/epoch - 2s/step
Epoch 107/150
33/33 - 63s - loss: 0.1005 - accuracy: 0.9708 - val_loss: 1.3407 - val_accuracy:
0.7376 - 63s/epoch - 2s/step
Epoch 108/150
33/33 - 64s - loss: 0.0726 - accuracy: 0.9796 - val_loss: 1.2361 - val_accuracy:
0.7414 - 64s/epoch - 2s/step
Epoch 109/150
33/33 - 63s - loss: 0.1003 - accuracy: 0.9679 - val_loss: 1.3945 - val_accuracy:
0.7414 - 63s/epoch - 2s/step
Epoch 110/150
33/33 - 63s - loss: 0.0776 - accuracy: 0.9757 - val_loss: 1.4965 - val_accuracy:
0.7338 - 63s/epoch - 2s/step
Epoch 111/150
33/33 - 63s - loss: 0.0897 - accuracy: 0.9727 - val_loss: 1.4180 - val_accuracy:
0.7338 - 63s/epoch - 2s/step
Epoch 112/150
33/33 - 63s - loss: 0.0731 - accuracy: 0.9757 - val_loss: 1.4165 - val_accuracy:
0.7186 - 63s/epoch - 2s/step
Epoch 113/150
```

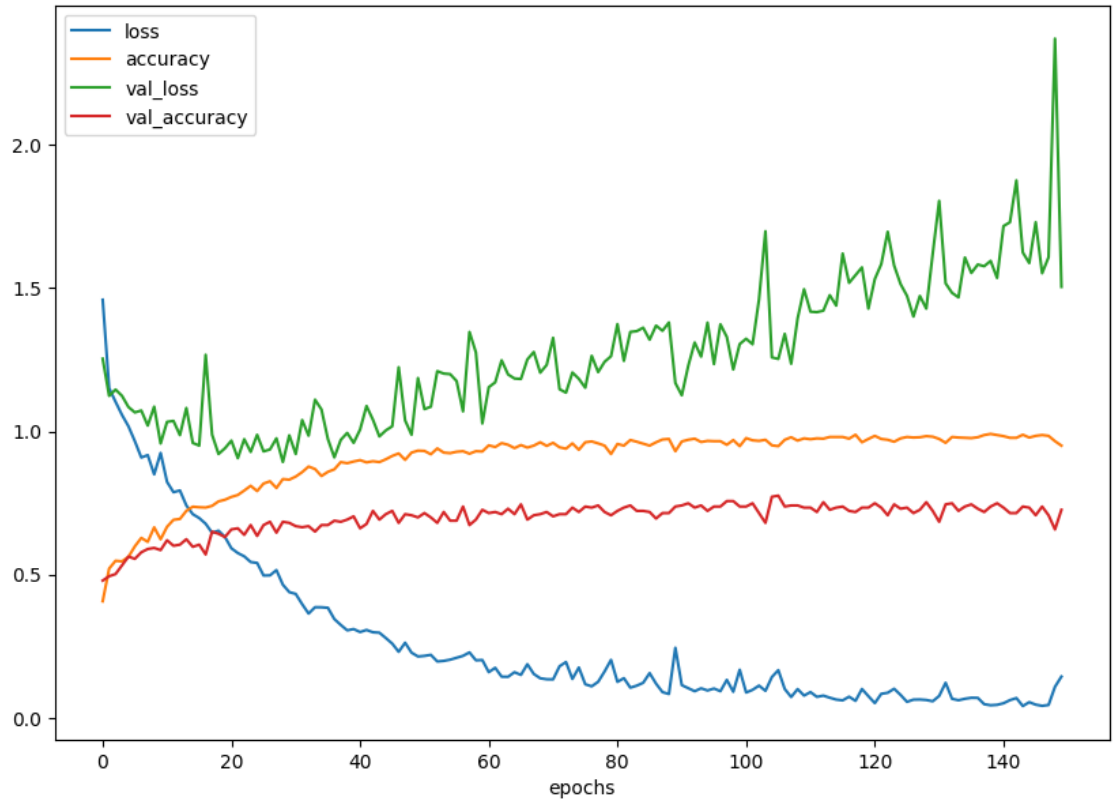
```
33/33 - 63s - loss: 0.0774 - accuracy: 0.9747 - val_loss: 1.4216 - val_accuracy:
0.7529 - 63s/epoch - 2s/step
Epoch 114/150
33/33 - 63s - loss: 0.0701 - accuracy: 0.9805 - val_loss: 1.4757 - val_accuracy:
0.7262 - 63s/epoch - 2s/step
Epoch 115/150
33/33 - 63s - loss: 0.0636 - accuracy: 0.9805 - val_loss: 1.4392 - val_accuracy:
0.7338 - 63s/epoch - 2s/step
Epoch 116/150
33/33 - 63s - loss: 0.0609 - accuracy: 0.9805 - val_loss: 1.6212 - val_accuracy:
0.7376 - 63s/epoch - 2s/step
Epoch 117/150
33/33 - 63s - loss: 0.0736 - accuracy: 0.9747 - val_loss: 1.5188 - val_accuracy:
0.7224 - 63s/epoch - 2s/step
Epoch 118/150
33/33 - 63s - loss: 0.0592 - accuracy: 0.9883 - val_loss: 1.5453 - val_accuracy:
0.7186 - 63s/epoch - 2s/step
Epoch 119/150
33/33 - 63s - loss: 0.1006 - accuracy: 0.9620 - val_loss: 1.5726 - val_accuracy:
0.7338 - 63s/epoch - 2s/step
Epoch 120/150
33/33 - 63s - loss: 0.0755 - accuracy: 0.9737 - val_loss: 1.4286 - val_accuracy:
0.7338 - 63s/epoch - 2s/step
Epoch 121/150
33/33 - 63s - loss: 0.0513 - accuracy: 0.9844 - val_loss: 1.5320 - val_accuracy:
0.7490 - 63s/epoch - 2s/step
Epoch 122/150
33/33 - 63s - loss: 0.0838 - accuracy: 0.9737 - val_loss: 1.5842 - val_accuracy:
0.7338 - 63s/epoch - 2s/step
Epoch 123/150
33/33 - 63s - loss: 0.0873 - accuracy: 0.9708 - val_loss: 1.6973 - val_accuracy:
0.7072 - 63s/epoch - 2s/step
Epoch 124/150
33/33 - 63s - loss: 0.1014 - accuracy: 0.9640 - val_loss: 1.5791 - val_accuracy:
0.7452 - 63s/epoch - 2s/step
Epoch 125/150
33/33 - 63s - loss: 0.0799 - accuracy: 0.9757 - val_loss: 1.5145 - val_accuracy:
0.7300 - 63s/epoch - 2s/step
Epoch 126/150
33/33 - 63s - loss: 0.0557 - accuracy: 0.9805 - val_loss: 1.4733 - val_accuracy:
0.7338 - 63s/epoch - 2s/step
Epoch 127/150
33/33 - 63s - loss: 0.0634 - accuracy: 0.9786 - val_loss: 1.4010 - val_accuracy:
0.7148 - 63s/epoch - 2s/step
Epoch 128/150
33/33 - 63s - loss: 0.0639 - accuracy: 0.9796 - val_loss: 1.4730 - val_accuracy:
0.7262 - 63s/epoch - 2s/step
Epoch 129/150
```

```
33/33 - 64s - loss: 0.0621 - accuracy: 0.9834 - val_loss: 1.4287 - val_accuracy:
0.7529 - 64s/epoch - 2s/step
Epoch 130/150
33/33 - 63s - loss: 0.0577 - accuracy: 0.9815 - val_loss: 1.6208 - val_accuracy:
0.7224 - 63s/epoch - 2s/step
Epoch 131/150
33/33 - 63s - loss: 0.0763 - accuracy: 0.9737 - val_loss: 1.8050 - val_accuracy:
0.6844 - 63s/epoch - 2s/step
Epoch 132/150
33/33 - 63s - loss: 0.1223 - accuracy: 0.9601 - val_loss: 1.5171 - val_accuracy:
0.7452 - 63s/epoch - 2s/step
Epoch 133/150
33/33 - 63s - loss: 0.0664 - accuracy: 0.9805 - val_loss: 1.4838 - val_accuracy:
0.7490 - 63s/epoch - 2s/step
Epoch 134/150
33/33 - 63s - loss: 0.0613 - accuracy: 0.9786 - val_loss: 1.4680 - val_accuracy:
0.7224 - 63s/epoch - 2s/step
Epoch 135/150
33/33 - 64s - loss: 0.0664 - accuracy: 0.9776 - val_loss: 1.6072 - val_accuracy:
0.7376 - 64s/epoch - 2s/step
Epoch 136/150
33/33 - 63s - loss: 0.0699 - accuracy: 0.9766 - val_loss: 1.5532 - val_accuracy:
0.7452 - 63s/epoch - 2s/step
Epoch 137/150
33/33 - 64s - loss: 0.0695 - accuracy: 0.9796 - val_loss: 1.5827 - val_accuracy:
0.7300 - 64s/epoch - 2s/step
Epoch 138/150
33/33 - 63s - loss: 0.0475 - accuracy: 0.9873 - val_loss: 1.5768 - val_accuracy:
0.7186 - 63s/epoch - 2s/step
Epoch 139/150
33/33 - 63s - loss: 0.0438 - accuracy: 0.9912 - val_loss: 1.5955 - val_accuracy:
0.7376 - 63s/epoch - 2s/step
Epoch 140/150
33/33 - 64s - loss: 0.0452 - accuracy: 0.9873 - val_loss: 1.5358 - val_accuracy:
0.7490 - 64s/epoch - 2s/step
Epoch 141/150
33/33 - 63s - loss: 0.0508 - accuracy: 0.9834 - val_loss: 1.7170 - val_accuracy:
0.7338 - 63s/epoch - 2s/step
Epoch 142/150
33/33 - 63s - loss: 0.0616 - accuracy: 0.9776 - val_loss: 1.7303 - val_accuracy:
0.7148 - 63s/epoch - 2s/step
Epoch 143/150
33/33 - 63s - loss: 0.0689 - accuracy: 0.9776 - val_loss: 1.8764 - val_accuracy:
0.7148 - 63s/epoch - 2s/step
Epoch 144/150
33/33 - 63s - loss: 0.0410 - accuracy: 0.9883 - val_loss: 1.6247 - val_accuracy:
0.7376 - 63s/epoch - 2s/step
Epoch 145/150
```

```
33/33 - 62s - loss: 0.0546 - accuracy: 0.9786 - val_loss: 1.5878 - val_accuracy:
0.7338 - 62s/epoch - 2s/step
Epoch 146/150
33/33 - 63s - loss: 0.0459 - accuracy: 0.9844 - val_loss: 1.7303 - val_accuracy:
0.7072 - 63s/epoch - 2s/step
Epoch 147/150
33/33 - 63s - loss: 0.0419 - accuracy: 0.9873 - val_loss: 1.5522 - val_accuracy:
0.7376 - 63s/epoch - 2s/step
Epoch 148/150
33/33 - 63s - loss: 0.0447 - accuracy: 0.9844 - val_loss: 1.6082 - val_accuracy:
0.7072 - 63s/epoch - 2s/step
Epoch 149/150
33/33 - 63s - loss: 0.1083 - accuracy: 0.9659 - val_loss: 2.3715 - val_accuracy:
0.6578 - 63s/epoch - 2s/step
Epoch 150/150
33/33 - 63s - loss: 0.1440 - accuracy: 0.9503 - val_loss: 1.5047 - val_accuracy:
0.7262 - 63s/epoch - 2s/step
```

[29]:
```
import pandas as pd
```

[30]:
```
pd.DataFrame(history.history).plot(figsize = (10, 7), xlabel = "epochs");
```



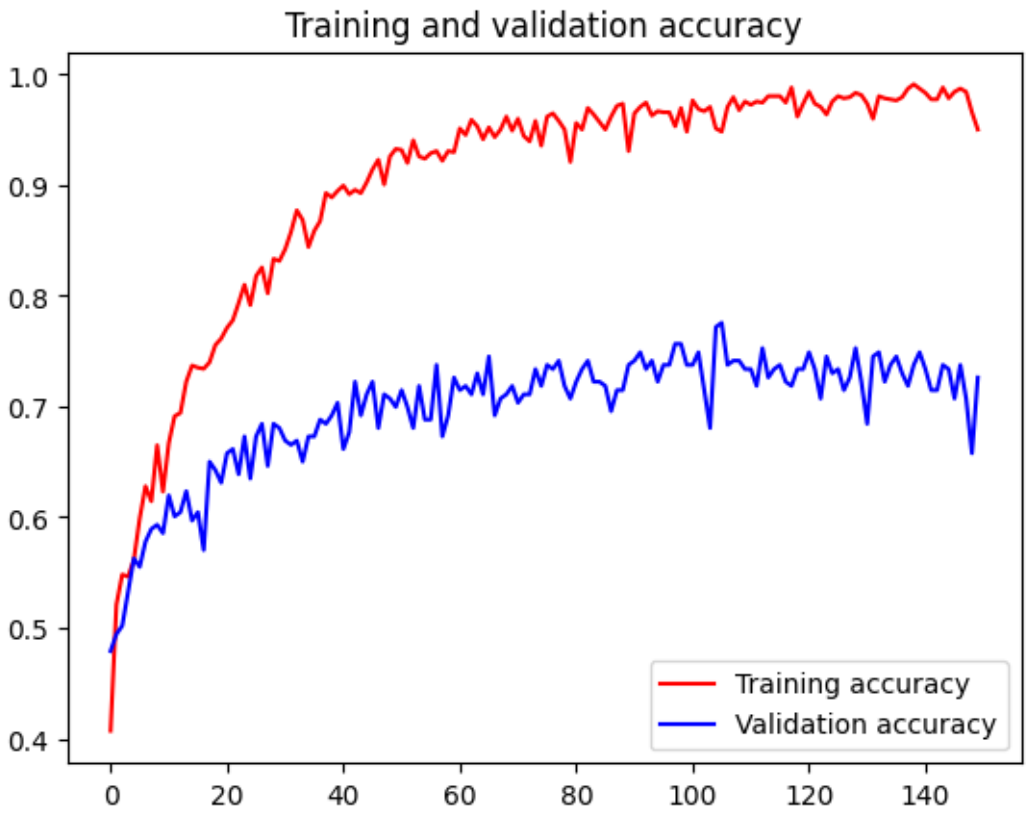
[31]:
```
# #plot the learning rate versus the loss
# lrs = 1e-4 * (10 ** (tf.range(300)/20))
# plt.figure(figsize = (10,7))
# plt.semilogx(lrs, history.history["loss"])
# plt.xlabel("learning rate")
# plt.ylabel("loss")
# plt.title("learning rate vs loss")
```

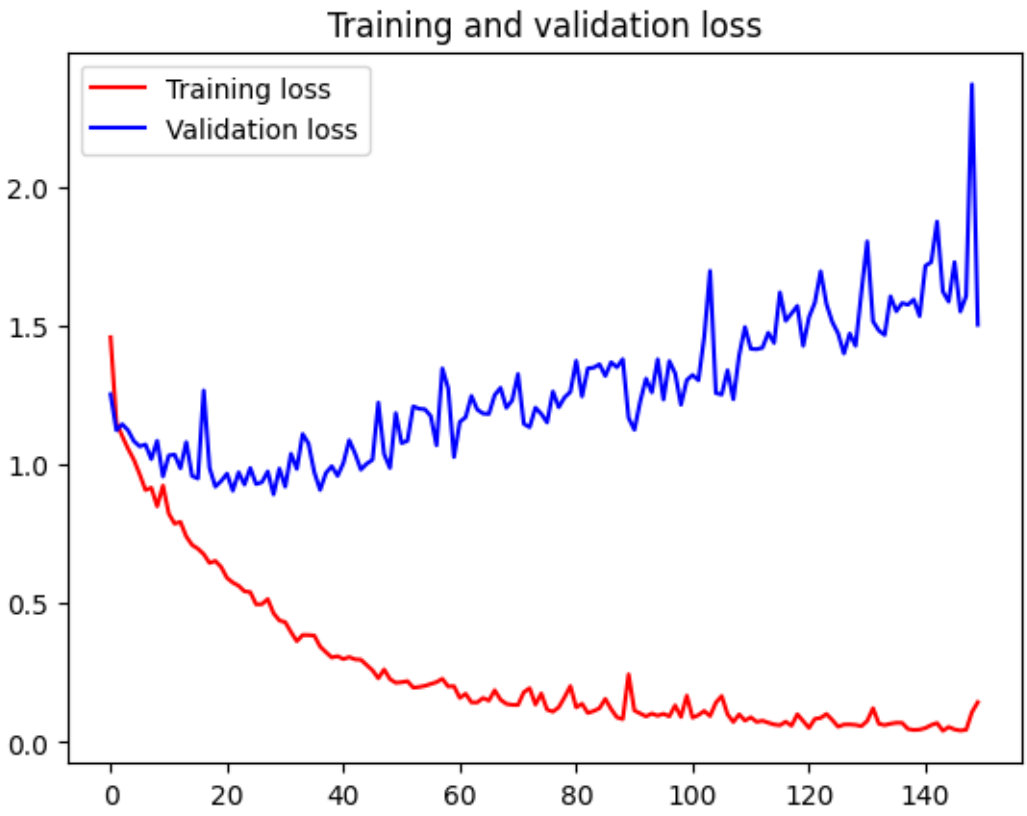
[32]:
```
%matplotlib inline

import matplotlib.image as mpimg
import matplotlib.pyplot as plt
```

[33]:
```
# evaluating accuracy and loss for the model
```

[34]:
```
#-----------------------------------------------------------
# Retrieve a list of list results on training and test data
# sets for each training epoch
#-----------------------------------------------------------
acc      = history.history[     'accuracy' ]
val_acc  = history.history[ 'val_accuracy' ]
loss     = history.history[    'loss' ]
val_loss = history.history['val_loss' ]

epochs   = range(len(acc)) # Get number of epochs

#------------------------------------------------
# Plot training and validation accuracy per epoch
#------------------------------------------------
plt.plot(epochs, acc, 'r', label='Training accuracy')
plt.plot(epochs, val_acc, 'b', label='Validation accuracy')
plt.title ('Training and validation accuracy')
plt.legend(loc=0)
plt.figure()

#------------------------------------------------
# Plot training and validation loss per epoch
#------------------------------------------------
plt.plot  ( epochs,     loss , 'r', label='Training loss')
plt.plot  ( epochs, val_loss , 'b', label='Validation loss')
plt.title ('Training and validation loss'   )
plt.legend(loc=0)
plt.figure()
```

[34]:
```
<Figure size 640x480 with 0 Axes>
```


Training and validation accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0
20
40
60
80
100
120
140
Training accuracy
Validation accuracy



<Figure size 640x480 with 0 Axes>

3 menyimpan model TAPI BELUM YANG VERSI QUANTIZED

[35]:
```
from keras.models import load_model

model.save('/home/jupyter/content/kerasFormat_model5/model_vggFace_13Juni.h5')
```

[36]:
```
from keras.models import load_model

model.save('/home/jupyter/content/saved Model5')
```

```
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op,
_jit_compiled_convolution_op, _jit_compiled_convolution_op,
_jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing
5 of 16). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: /home/jupyter/content/saved Model5/assets

INFO:tensorflow:Assets written to: /home/jupyter/content/saved Model5/assets
```

4 Convert to Tensorflow LITE

[ ]:
```
try:
    %tensorflow_version 2.x
except:
    pass
```

[ ]:
```
import pathlib

print('\u2022 Using TensorFlow Version:', tf.__version__)
```

[ ]:
```

```

[ ]:
```
#!mkdir saved_TFLITE_model
```

generate SavedModel

[ ]:
```
export_dir = "home/jupyter/content/saved_TFLITE_model"
tf.saved_model.save(model, export_dir)
```

[ ]:
```
model = tf.saved_model.load(export_dir)
```

Convert the SavedModel to JSON

[ ]:
```
#!pip install --user tensorflowjs
```

[ ]:
```
# import tensorflowjs
# import json
```

[ ]:
```
# # Load the SavedModel
# model = tf.saved_model.load('home/jupyter/content/saved Model')
```

[ ]:
```
import tensorflow as tf
import json

# Load the SavedModel
model = tf.saved_model.load(export_dir)

# Convert tensor shapes to lists
def convert_shape(shape):
    return [dim for dim in shape.as_list()]

# Create a dictionary to store the JSON model
json_model = {}

# Get information about inputs
input_signatures = model.signatures['serving_default'].
 ↪structured_input_signature[1]
json_model['inputs'] = []
```

```
for tensor_name, tensor_info in input_signatures.items():
    input_info = {
        'name': tensor_name,
        'dtype': str(tensor_info.dtype),
        'shape': convert_shape(tensor_info.shape),
    }
    json_model['inputs'].append(input_info)

# Get information about outputs
output_signatures = model.signatures['serving_default'].structured_outputs
json_model['outputs'] = []
for tensor_name, tensor_info in output_signatures.items():
    output_info = {
        'name': tensor_name,
        'dtype': str(tensor_info.dtype),
        'shape': convert_shape(tensor_info.shape),
    }
    json_model['outputs'].append(output_info)

# Save the JSON model to a file
with open('model.json', 'w') as f:
    json.dump(json_model, f)
```

Convert the SavedModel to TFLite

[ ]:
```
# # Convert the model.
# converter = tf.lite.TFLiteConverter.from_saved_model(export_dir)

# # Set the optimization flags for quantization
# converter.optimizations = [tf.lite.Optimize.DEFAULT] # INTINYA DISINI UNTUK␣
  ↪MELAKUKAN QUANTIZED

# # Specify the input and output tensors (if necessary)
# # converter.target_spec.supported_ops = [tf.lite.OpsSet.TFLITE_BUILTINS_INT8]
# # converter.target_spec.supported_types = [tf.int8]

# tflite_model = converter.convert()
```

[ ]:
```
# tflite_model_file = pathlib.Path('home/jupyter/content/')
# tflite_model_file.write_bytes(tflite_model)
```

5 labels

[ ]:
```
class_names = ['combination', 'dry', 'nonface' ,'normal', 'oily', 'sensitive']
```

create a txt file to save the 5 labels

[ ]:
```
with open('faces_labels.txt', 'w') as f:
    f.write('\n'.join(class_names))
```

[ ]:
```
import shutil
shutil.copy("/content/faces_labels.txt","/content/drive/MyDrive/saved model")
```

[ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

[ ]:
```
import os
file_path = "/home/jupyter/content/kerasFormat_model/model_vggFace_10Juni.h5" ␣
 ↪# Replace with the correct file path
file_size = os.path.getsize(file_path)
print(file_size)
```

[ ]:
```
!wget -O /home/jupyter/content/kerasFormat_model/model_vggFace_10Juni.h5 https:/
 ↪/58948e97954de7e0-dot-asia-southeast2.notebooks.googleusercontent.com/files/
 ↪content/kerasFormat_model/model_vggFace_10Juni.h5?
 ↪_xsrf=2%7C1f091a48%7C9c97e6fd85cef3a278dc9a8e0d4188fa%7C1686143743
```

[ ]:
```

```