Research Paper Implementation

November 2, 2023

[2]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

import cv2
import os
os.environ["TF_CPP_MIN_LOG_LEVEL"] = "2"
import warnings
warnings.filterwarnings('ignore')

from sklearn.metrics import confusion_matrix, classification_report

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Activation, BatchNormalization, Conv2D,␣
↪Dense, Dropout, Flatten, MaxPooling2D
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.losses import CategoricalCrossentropy
from tensorflow.keras.regularizers import l2
from tensorflow.keras.callbacks import ReduceLROnPlateau, EarlyStopping
```

[3]:
```
import pandas as pd
import numpy as np
import itertools
import keras
from sklearn import metrics
from sklearn.metrics import confusion_matrix
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras import optimizers
from keras.preprocessing import image
from keras.layers import Dropout, Flatten, Dense
from keras import applications
from keras.utils.np_utils import to_categorical
import matplotlib.pyplot as plt
```

```
import matplotlib.image as mpimg
%matplotlib inline
import math
import datetime
import time
from tensorflow.keras.utils import img_to_array,load_img
```

[4]:
```
tf.config.experimental_run_functions_eagerly(True)
```

```
WARNING:tensorflow:From
C:\Users\91797\AppData\Local\Temp\ipykernel_15768\2874510872.py:1:
experimental_run_functions_eagerly (from
tensorflow.python.eager.polymorphic_function.quarantine) is deprecated and will
be removed in a future version.
Instructions for updating:
Use `tf.config.run_functions_eagerly` instead of the experimental version.
```

[5]:
```
tf.config.run_functions_eagerly(True)
```

[6]:
```
pip install opencv-python
```

```
Requirement already satisfied: opencv-python in
c:\users\91797\appdata\local\programs\python\python310\lib\site-packages
(4.8.1.78)
Requirement already satisfied: numpy>=1.21.2 in
c:\users\91797\appdata\local\programs\python\python310\lib\site-packages (from
opencv-python) (1.23.5)
Note: you may need to restart the kernel to use updated packages.
```

[7]:
```
pip install seaborn
```

```
Requirement already satisfied: seaborn in
c:\users\91797\appdata\local\programs\python\python310\lib\site-packages
(0.13.0)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in
c:\users\91797\appdata\local\programs\python\python310\lib\site-packages (from
seaborn) (1.23.5)
Requirement already satisfied: pandas>=1.2 in
c:\users\91797\appdata\local\programs\python\python310\lib\site-packages (from
seaborn) (2.1.2)
Requirement already satisfied: matplotlib!=3.6.1,>=3.3 in
c:\users\91797\appdata\local\programs\python\python310\lib\site-packages (from
seaborn) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\91797\appdata\local\programs\python\python310\lib\site-packages (from
matplotlib!=3.6.1,>=3.3->seaborn) (1.0.7)
Requirement already satisfied: cycler>=0.10 in
c:\users\91797\appdata\local\programs\python\python310\lib\site-packages (from
```

```
matplotlib!=3.6.1,>=3.3->seaborn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\91797\appdata\local\programs\python\python310\lib\site-packages (from
matplotlib!=3.6.1,>=3.3->seaborn) (4.39.3)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\users\91797\appdata\local\programs\python\python310\lib\site-packages (from
matplotlib!=3.6.1,>=3.3->seaborn) (1.4.4)
Requirement already satisfied: packaging>=20.0 in
c:\users\91797\appdata\local\programs\python\python310\lib\site-packages (from
matplotlib!=3.6.1,>=3.3->seaborn) (23.1)
Requirement already satisfied: pillow>=6.2.0 in
c:\users\91797\appdata\local\programs\python\python310\lib\site-packages (from
matplotlib!=3.6.1,>=3.3->seaborn) (9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in
c:\users\91797\appdata\local\programs\python\python310\lib\site-packages (from
matplotlib!=3.6.1,>=3.3->seaborn) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\91797\appdata\local\programs\python\python310\lib\site-packages (from
matplotlib!=3.6.1,>=3.3->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
c:\users\91797\appdata\local\programs\python\python310\lib\site-packages (from
pandas>=1.2->seaborn) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in
c:\users\91797\appdata\local\programs\python\python310\lib\site-packages (from
pandas>=1.2->seaborn) (2023.3)
Requirement already satisfied: six>=1.5 in
c:\users\91797\appdata\local\programs\python\python310\lib\site-packages (from
python-dateutil>=2.7->matplotlib!=3.6.1,>=3.3->seaborn) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

[8]:
```
train_data_dir = 'C:\\Users\\91797\\Untitled Folder 1\\actual data\\train'
validation_data_dir = 'C:\\Users\\91797\\Untitled Folder 1\\actual data\\test'
```

[10]:
```
Train = keras.utils.image_dataset_from_directory(
    directory=train_data_dir,
    labels="inferred",
    label_mode="categorical",
    batch_size=32,
    image_size=(256, 256))
Validation = keras.utils.image_dataset_from_directory(
    directory=validation_data_dir,
    labels="inferred",
    label_mode="categorical",
    batch_size=32,
    image_size=(256, 256))
```

```
Found 5232 files belonging to 2 classes.
Found 624 files belonging to 2 classes.
```

[11]:
```
from tensorflow.keras import layers
```

[12]:
```
model = tf.keras.models.Sequential([
    layers.Conv2D(64, (3, 3), activation='relu', input_shape=(256, 256, 3)),
    layers.BatchNormalization(),
    layers.MaxPooling2D(2, 2),
    layers.Conv2D(128, (3, 3), activation='relu'),
    layers.Dropout(0.2),
    layers.BatchNormalization(),
    layers.MaxPooling2D(2, 2),
    layers.Conv2D(128, (3, 3), activation='relu'),
     layers.BatchNormalization(),
    layers.MaxPooling2D(2, 2),
    layers.Conv2D(256, (3, 3), activation='relu'),
     layers.Dropout(0.2),
    layers.BatchNormalization(),
    layers.MaxPooling2D(2, 2),
 layers.Conv2D(512, (3, 3), activation='relu'),
    layers.Dropout(0.2),
     layers.BatchNormalization(),
    layers.MaxPooling2D(2, 2),
    layers.Flatten(),
    layers.Dense(256, activation='relu'),
    layers.Dropout(0.1),
    layers.Dense(2, activation='sigmoid')
])
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 254, 254, 64)      1792

 batch_normalization (BatchN  (None, 254, 254, 64)     256
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 127, 127, 64)     0
 )

 conv2d_1 (Conv2D)           (None, 125, 125, 128)     73856

 dropout (Dropout)           (None, 125, 125, 128)     0

 batch_normalization_1 (Batc  (None, 125, 125, 128)    512
 hNormalization)
```

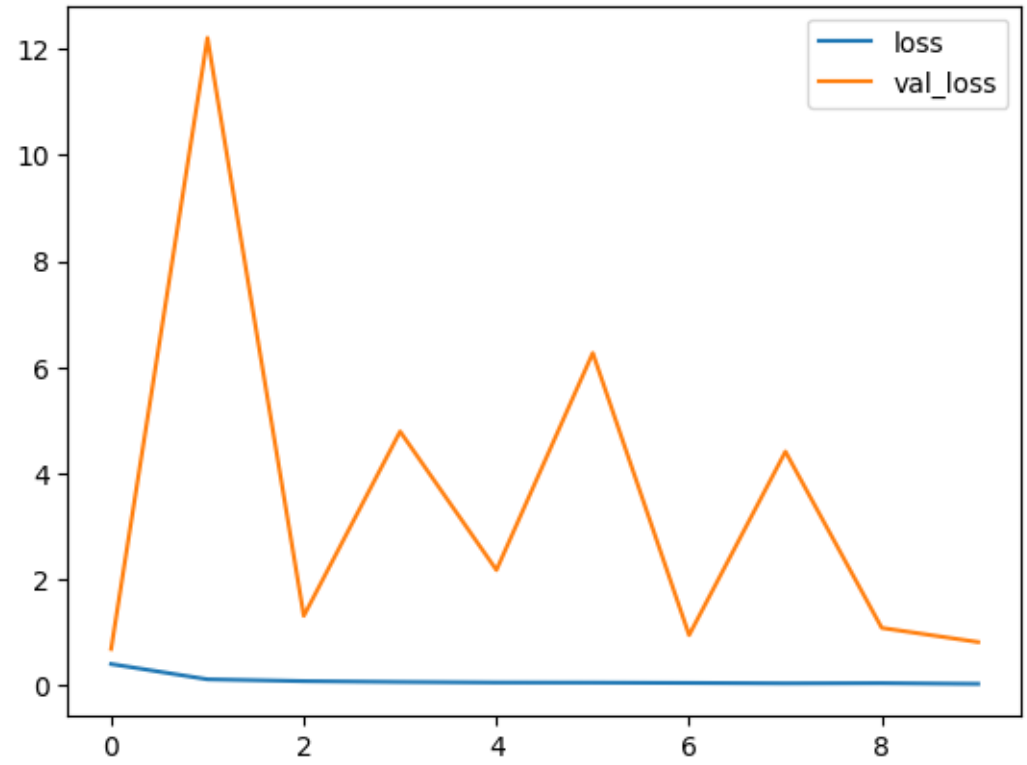
```
max_pooling2d_1 (MaxPooling  (None, 62, 62, 128)      0
2D)

conv2d_2 (Conv2D)            (None, 60, 60, 128)       147584

batch_normalization_2 (Batc  (None, 60, 60, 128)      512
hNormalization)

max_pooling2d_2 (MaxPooling  (None, 30, 30, 128)      0
2D)

conv2d_3 (Conv2D)            (None, 28, 28, 256)       295168

dropout_1 (Dropout)          (None, 28, 28, 256)       0

batch_normalization_3 (Batc  (None, 28, 28, 256)      1024
hNormalization)

max_pooling2d_3 (MaxPooling  (None, 14, 14, 256)      0
2D)

conv2d_4 (Conv2D)            (None, 12, 12, 512)       1180160

dropout_2 (Dropout)          (None, 12, 12, 512)       0

batch_normalization_4 (Batc  (None, 12, 12, 512)      2048
hNormalization)

max_pooling2d_4 (MaxPooling  (None, 6, 6, 512)        0
2D)

flatten (Flatten)            (None, 18432)             0

dense (Dense)                (None, 256)               4718848

dropout_3 (Dropout)          (None, 256)               0

dense_1 (Dense)              (None, 2)                 514

=================================================================
Total params: 6,422,274
Trainable params: 6,420,098
Non-trainable params: 2,176
_________________________________________________________________
```

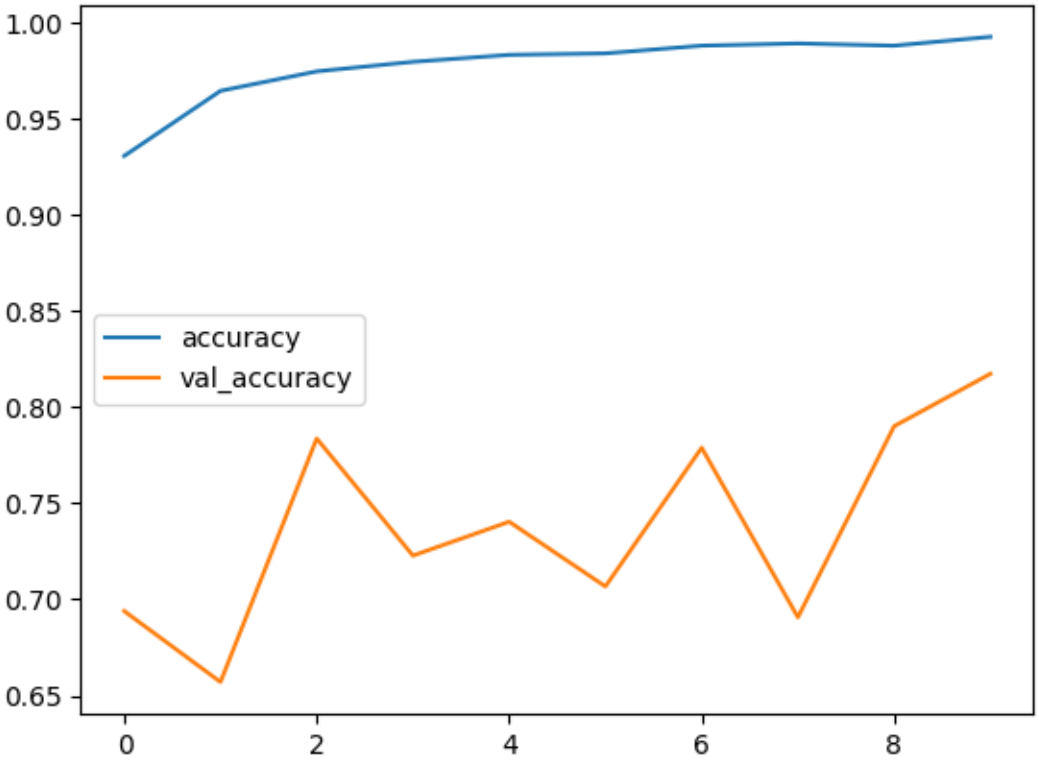
[13]:
```
model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
```

[14]:
```
history = model.fit(Train,
          epochs=10,
          validation_data=Validation)
```

```
Epoch 1/10
164/164 [==============================] - 1727s 11s/step - loss: 0.3955 -
accuracy: 0.9306 - val_loss: 0.6815 - val_accuracy: 0.6939
Epoch 2/10
164/164 [==============================] - 1737s 11s/step - loss: 0.1072 -
accuracy: 0.9644 - val_loss: 12.2111 - val_accuracy: 0.6571
Epoch 3/10
164/164 [==============================] - 1729s 11s/step - loss: 0.0729 -
accuracy: 0.9746 - val_loss: 1.3046 - val_accuracy: 0.7837
Epoch 4/10
164/164 [==============================] - 1739s 11s/step - loss: 0.0578 -
accuracy: 0.9795 - val_loss: 4.7840 - val_accuracy: 0.7228
Epoch 5/10
164/164 [==============================] - 1739s 11s/step - loss: 0.0469 -
accuracy: 0.9832 - val_loss: 2.1694 - val_accuracy: 0.7404
Epoch 6/10
164/164 [==============================] - 1500s 9s/step - loss: 0.0448 -
accuracy: 0.9839 - val_loss: 6.2652 - val_accuracy: 0.7067
Epoch 7/10
164/164 [==============================] - 1364s 8s/step - loss: 0.0389 -
accuracy: 0.9880 - val_loss: 0.9416 - val_accuracy: 0.7788
Epoch 8/10
164/164 [==============================] - 1372s 8s/step - loss: 0.0319 -
accuracy: 0.9891 - val_loss: 4.4043 - val_accuracy: 0.6907
Epoch 9/10
164/164 [==============================] - 1584s 10s/step - loss: 0.0351 -
accuracy: 0.9880 - val_loss: 1.0771 - val_accuracy: 0.7901
Epoch 10/10
164/164 [==============================] - 1576s 10s/step - loss: 0.0215 -
accuracy: 0.9925 - val_loss: 0.8103 - val_accuracy: 0.8173
```

[18]:
```
history_df = pd.DataFrame(history.history)
history_df.loc[:, ['loss', 'val_loss']].plot()
history_df.loc[:, ['accuracy', 'val_accuracy']].plot()
plt.show()
```


loss
val_loss
12
10
8
6
4
2
0
0
2
4
6
8


accuracy
val_accuracy
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0
2
4
6
8