

Lampiran 1

Source Code

```
{
  "cells": [
    {
      "cell_type": "code",
      "execution_count": 1,
      "metadata": {},
      "outputs": [
        {
          "name": "stderr",
          "output_type": "stream",
          "text": "Using TensorFlow backend.\n"
        }
      ],
      "source": "from keras.models import Sequential\nfrom keras.\n    layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout,\n    Activation, BatchNormalization"
    },
    {
      "cell_type": "code",
      "execution_count": 2,
      "metadata": {},
      "outputs": [],
      "source": "import numpy as np\nimport pandas as pd\nimport\n    matplotlib.pyplot as plt\nimport tensorflow\n%matplotlib\n    inline"
    },
    {
      "cell_type": "code",
      "execution_count": 3,
      "metadata": {},
      "outputs": [
        {
          "name": "stdout",
          "output_type": "stream",

```

```

"text": "Model: \"sequential_1\"\\n
-----\\n
Layer (type)                 Output Shape
Param #      \\n
-----\\n
nconv2d_1 (Conv2D)           (None, 198, 198, 32)
896          \\n
-----\\n
nconv2d_2 (Conv2D)           (None, 196, 196, 16)
4624         \\n
-----\\n
nmax_pooling2d_1 (MaxPooling2 (None, 65, 65, 16)      0
              \\n
-----\\n
nflatten_1 (Flatten)         (None, 67600)      0
              \\n
-----\\n
ndense_1 (Dense)             (None, 12)
811212        \\n
-----\\n
ndropout_1 (Dropout)         (None, 12)         0
              \\n
-----\\n
ndense_2 (Dense)             (None, 24)
312           \\n
-----\\n
nTotal params: 817,044\\nTrainable params: 817,044\\nNon-
trainable params: 0\\n
-----\\n
\\n"
}
],
"source": "nb_classes = 24\\nmodel = Sequential()\\nmodel.add(
    Conv2D(32, kernel_size=(3,3), activation='relu', input_shape
    =(200,200,3))\\nmodel.add(Conv2D(16, kernel_size=(3,3),
    activation='relu', input_shape=(200,200,3)))\\nmodel.add(
    MaxPooling2D(pool_size=(3, 3)))\\nmodel.add(Flatten())\\nmodel
    .add(Dense(12, activation= 'relu'))\\nmodel.add(Dropout(0.2))
    \\nmodel.add(Dense(nb_classes, activation = 'softmax'))\\n#
    model.layers[0].trainable = True\\nmodel.summary()"
},
{
    "cell_type": "code",
    "execution_count": 8,
    "metadata": {},

```

```

"outputs": [],
"source": "from keras.optimizers import RMSprop\noptimizers =
    RMSprop(lr=0.00001)\nmodel.compile(\n    optimizer=
    optimizers,\n    loss='categorical_crossentropy',\n    metrics=['accuracy'])\n",
},
{
"cell_type": "code",
"execution_count": 9,
"metadata": {},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": "Found 1712 images belonging to 24 classes.\nFound
    318 images belonging to 24 classes.\n"
}
],
"source": "from keras.preprocessing.image import
    ImageDataGenerator\nfrom keras.models import Sequential\n
from keras.layers import Dropout, Flatten, Dense\nfrom
keras import applications\ntrain_datagen =
    ImageDataGenerator(rescale=1./255,\n
                        horizontal_flip=False)\n
    test_datagen = ImageDataGenerator(rescale=1./255)\nnclasses
= ['A', 'A#', 'A#m', 'Am', 'B', 'Bm',\n    'C', 'C#', 'C#m',
    'Cm', 'D', 'D#',\n    'D#m', 'Dm', 'E', 'Em', 'F', 'F#',
    'F#m', 'Fm', 'G', 'G#', 'G#m', 'Gm']\n
train_generator = train_datagen.flow_from_directory(\n
    'train/data_train',\n    target_size=(200, 200),\n
    shuffle = True,\n    batch_size=64,\n
    class_mode='categorical')\ntest_generator =
    test_datagen.flow_from_directory(\n
    'train/data_test',\n    target_size=(200, 200),\n
    shuffle = True,\n    batch_size=247,\n
    class_mode='categorical')
",
},
{
"cell_type": "code",
"execution_count": 10,
"metadata": {},
"outputs": [],
"source": "from keras.callbacks import ModelCheckpoint\n
    filepath='weight.checkpoint5.cont.h5'\ncheckpoint = [
    ModelCheckpoint(filepath, monitor='val_loss', verbose=1,

```

```

        save_best_only=True, mode='min'),\n
        ModelCheckpoint(filepath, monitor='loss', verbose=1,\n
        save_best_only=True, mode='min')]"
    },
    {
        "cell_type": "code",
        "execution_count": 11,
        "metadata": {
            "scrolled": true
        },
        "outputs": [
            {
                "name": "stdout",
                "output_type": "stream",
                "text": "Epoch 1/1000\n26/26 [=====]\n
                    - 60s 2s/step - loss: 3.0627 - accuracy: 0.1438 -\n
                    val_loss: 3.1323 - val_accuracy: 0.0409\n\nEpoch 00001:\n
                    val_loss improved from inf to 3.13228, saving model to\n
                    weight.checkpoint5.cont.h5\n\nEpoch 00001: loss improved\n
                    from inf to 3.06291, saving model to weight.checkpoint5.\n
                    cont.h5\nEpoch 2/1000\n26/26\n
                    [=====] - 57s 2s/step - loss:\n
                    3.0615 - accuracy: 0.1456 - val_loss: 3.2120 -\n
                    val_accuracy: 0.0409\n\nEpoch 00002: val_loss did not\n
                    improve from 3.13228\n\nEpoch 00002: loss improved from\n
                    3.06291 to 3.06162, saving model to weight.checkpoint5.\n
                    cont.h5\nEpoch 3/1000\n26/26\n
                    [=====] - 56s 2s/step - loss:\n
                    3.0638 - accuracy: 0.1468 - val_loss: 3.2062 -\n
                    val_accuracy: 0.0409\n\nEpoch 00003: val_loss did not\n
                    improve from 3.13228\n\nEpoch 00003: loss did not improve\n
                    from 3.06162\nEpoch 4/1000\n26/26\n
                    [=====] - 71s 3s/step - loss:\n
                    3.0567 - accuracy: 0.1462 - val_loss: 3.1539 -\n
                    val_accuracy: 0.0409\n\nEpoch 00004: val_loss did not\n
                    improve from 3.13228\n\nEpoch 00004: loss improved from\n
                    3.06162 to 3.05760, saving model to weight.checkpoint5.\n
                    cont.h5\nEpoch 5/1000\n26/26\n
                    [=====] - 56s 2s/step - loss:\n
                    3.0700 - accuracy: 0.1390 - val_loss: 3.1996 -\n
                    val_accuracy: 0.0409\n\nEpoch 00005: val_loss did not\n
                    improve from 3.13228\n\nEpoch 00005: loss did not improve\n
                    from 3.05760\nEpoch 6/1000\n26/26\n
                    [=====] - 78s 3s/step - loss:\n
                    3.0613 - accuracy: 0.1448 - val_loss: 3.1987 -

```

```

val_accuracy: 0.0409\n\nEpoch 00006: val_loss did not
improve from 3.13228\n\nEpoch 00006: loss did not improve
from 3.05760\nEpoch 7/1000\n26/26
[=====] - 76s 3s/step - loss:
3.0699 - accuracy: 0.1379 - val_loss: 3.1736 -
val_accuracy: 0.0409\n\nEpoch 00007: val_loss did not
improve from 3.13228\n\nEpoch 00007: loss did not improve
from 3.05760\nEpoch 8/1000\n26/26
[=====] - 62s 2s/step - loss:
3.0619 - accuracy: 0.1490 - val_loss: 3.1940 -
val_accuracy: 0.0409\n\nEpoch 00008: val_loss did not
improve from 3.13228\n\nEpoch 00008: loss did not improve
from 3.05760\nEpoch 9/1000\n26/26
[=====] - 56s 2s/step - loss:
3.0475 - accuracy: 0.1566 - val_loss: 3.1843 -
val_accuracy: 0.0409\n\nEpoch 00009: val_loss did not
improve from 3.13228\n\nEpoch 00009: loss improved from
3.05760 to 3.04834, saving model to weight.checkpoint5.
cont.h5\nEpoch 10/1000\n26/26
[=====] - 56s 2s/step - loss:
3.0682 - accuracy: 0.1403 - val_loss: 3.2236 -
val_accuracy: 0.0409\n\nEpoch 00010: val_loss did not
improve from 3.13228\n\nEpoch 00010: loss did not improve
from 3.04834\nEpoch 11/1000\n26/26
[=====] - 56s 2s/step - loss:
3.0626 - accuracy: 0.1426 - val_loss: 3.1664 -
val_accuracy: 0.0409\n\nEpoch 00011: val_loss did not
improve from 3.13228\n\nEpoch 00011: loss did not improve
from 3.04834\nEpoch 12/1000\n26/26
[=====] - 55s 2s/step - loss:
3.0682 - accuracy: 0.1424 - val_loss: 3.1899 -
val_accuracy: 0.0409\n\nEpoch 00012: val_loss did not
improve from 3.13228\n\nEpoch 00012: loss did not improve
from 3.04834\nEpoch 13/1000\n26/26
[=====] - 61s 2s/step - loss:
3.0583 - accuracy: 0.1499 - val_loss: 3.2304 -
val_accuracy: 0.0409\n\nEpoch 00013: val_loss did not
improve from 3.13228\n\nEpoch 00013: loss did not improve
from 3.04834\nEpoch 14/1000\n26/26
[=====] - 66s 3s/step - loss:
3.0549 - accuracy: 0.1477 - val_loss: 3.2015 -
val_accuracy: 0.0409\n\nEpoch 00014: val_loss did not
improve from 3.13228\n\nEpoch 00014: loss did not improve
from 3.04834\nEpoch 15/1000\n26/26
[=====] - 67s 3s/step - loss:

```

```

3.0755 - accuracy: 0.1370 - val_loss: 3.1619 -
val_accuracy: 0.0409\n\nEpoch 00015: val_loss did not
improve from 3.13228\n\nEpoch 00015: loss did not improve
from 3.04834\nEpoch 16/1000\n26/26
[=====] - 66s 3s/step - loss:
3.0631 - accuracy: 0.1446 - val_loss: 3.1893 -
val_accuracy: 0.0409\n\nEpoch 00016: val_loss did not
improve from 3.13228\n\nEpoch 00016: loss did not improve
from 3.04834\nEpoch 17/1000\n26/26
[=====] - 67s 3s/step - loss:
3.0558 - accuracy: 0.1496 - val_loss: 3.2430 -
val_accuracy: 0.0409\n\nEpoch 00017: val_loss did not
improve from 3.13228\n\nEpoch 00017: loss did not improve
from 3.04834\nEpoch 18/1000\n26/26
[=====] - 61s 2s/step - loss:
3.0616 - accuracy: 0.1471 - val_loss: 3.1461 -
val_accuracy: 0.0409\n\nEpoch 00018: val_loss did not
improve from 3.13228\n\nEpoch 00018: loss did not improve
from 3.04834\nEpoch 19/1000\n26/26
[=====] - 659s 25s/step - loss:
3.0645 - accuracy: 0.1420 - val_loss: 3.2351 -
val_accuracy: 0.0409\n\nEpoch 00019: val_loss did not
improve from 3.13228\n\nEpoch 00019: loss did not improve
from 3.04834\nEpoch 20/1000\n26/26
[=====] - 54s 2s/step - loss:
3.0592 - accuracy: 0.1444 - val_loss: 3.1370 -
val_accuracy: 0.0409\n\nEpoch 00020: val_loss did not
improve from 3.13228\n\nEpoch 00020: loss did not improve
from 3.04834\nEpoch 21/1000\n26/26
[=====] - 80s 3s/step - loss:
3.0646 - accuracy: 0.1472 - val_loss: 3.1770 -
val_accuracy: 0.0409\n\nEpoch 00021: val_loss did not
improve from 3.13228\n\nEpoch 00021: loss did not improve
from 3.04834\nEpoch 22/1000\n26/26
[=====] - 64s 2s/step - loss:
3.0630 - accuracy: 0.1475 - val_loss: 3.2277 -
val_accuracy: 0.0409\n\nEpoch 00022: val_loss did not
improve from 3.13228\n\nEpoch 00022: loss did not improve
from 3.04834\nEpoch 23/1000\n26/26
[=====] - 55s 2s/step - loss:
3.0635 - accuracy: 0.1408 - val_loss: 3.1488 -
val_accuracy: 0.0409\n\nEpoch 00023: val_loss did not
improve from 3.13228\n\nEpoch 00023: loss did not improve
from 3.04834\nEpoch 24/1000\n26/26
[=====] - 70s 3s/step - loss:

```

```

3.0685 - accuracy: 0.1409 - val_loss: 3.1949 -
val_accuracy: 0.0409\n\nEpoch 00024: val_loss did not
improve from 3.13228\n\nEpoch 00024: loss did not improve
from 3.04834\nEpoch 25/1000\n26/26
[=====] - 65s 3s/step - loss:
3.0628 - accuracy: 0.1472 - val_loss: 3.1538 -
val_accuracy: 0.0409\n\nEpoch 00025: val_loss did not
improve from 3.13228\n\nEpoch 00025: loss did not improve
from 3.04834\nEpoch 26/1000\n26/26
[=====] - 66s 3s/step - loss:
3.0594 - accuracy: 0.1456 - val_loss: 3.2175 -
val_accuracy: 0.0409\n\nEpoch 00026: val_loss did not
improve from 3.13228\n\nEpoch 00026: loss did not improve
from 3.04834\nEpoch 27/1000\n26/26
[=====] - 65s 2s/step - loss:
3.0626 - accuracy: 0.1444 - val_loss: 3.2237 -
val_accuracy: 0.0409\n\nEpoch 00027: val_loss did not
improve from 3.13228\n\nEpoch 00027: loss did not improve
from 3.04834\nEpoch 28/1000\n26/26
[=====] - 63s 2s/step - loss:
3.0645 - accuracy: 0.1432 - val_loss: 3.2014 -
val_accuracy: 0.0409\n\nEpoch 00028: val_loss did not
improve from 3.13228\n\nEpoch 00028: loss did not improve
from 3.04834\nEpoch 29/1000\n26/26
[=====] - 59s 2s/step - loss:
3.0645 - accuracy: 0.1438 - val_loss: 3.2313 -
val_accuracy: 0.0409\n\nEpoch 00029: val_loss did not
improve from 3.13228\n\nEpoch 00029: loss did not improve
from 3.04834\nEpoch 30/1000\n26/26
[=====] - 54s 2s/step - loss:
3.0619 - accuracy: 0.1468 - val_loss: 3.2027 -
val_accuracy: 0.0409\n\nEpoch 00030: val_loss did not
improve from 3.13228\n\nEpoch 00030: loss did not improve
from 3.04834\nEpoch 31/1000\n26/26
[=====] - 50s 2s/step - loss:
3.0614 - accuracy: 0.1438 - val_loss: 3.1087 -
val_accuracy: 0.0409\n\nEpoch 00031: val_loss improved
from 3.13228 to 3.10871, saving model to weight.
checkpoint5.cont.h5\n\nEpoch 00031: loss did not improve
from 3.04834\nEpoch 32/1000\n26/26
[=====] - 51s 2s/step - loss:
3.0640 - accuracy: 0.1450 - val_loss: 3.1754 -
val_accuracy: 0.0409\n\nEpoch 00032: val_loss did not
improve from 3.10871\n\nEpoch 00032: loss did not improve
from 3.04834\nEpoch 33/1000\n26/26

```

```

=====] - 51s 2s/step - loss:
3.0563 - accuracy: 0.1475 - val_loss: 3.2294 -
val_accuracy: 0.0409\n\nEpoch 00033: val_loss did not
improve from 3.10871\n\nEpoch 00033: loss did not improve
from 3.04834\nEpoch 34/1000\n"
},
{
"name": "stdout",
"output_type": "stream",
"text": "26/26 =====] - 59s 2s/step
- loss: 3.0678 - accuracy: 0.1408 - val_loss: 3.1755 -
val_accuracy: 0.0409\n\nEpoch 00034: val_loss did not
improve from 3.10871\n\nEpoch 00034: loss did not improve
from 3.04834\nEpoch 35/1000\n26/26
=====] - 54s 2s/step - loss:
3.0665 - accuracy: 0.1481 - val_loss: 3.1536 -
val_accuracy: 0.0409\n\nEpoch 00035: val_loss did not
improve from 3.10871\n\nEpoch 00035: loss did not improve
from 3.04834\nEpoch 36/1000\n26/26
=====] - 57s 2s/step - loss:
3.0547 - accuracy: 0.1514 - val_loss: 3.1829 -
val_accuracy: 0.0409\n\nEpoch 00036: val_loss did not
improve from 3.10871\n\nEpoch 00036: loss did not improve
from 3.04834\nEpoch 37/1000\n26/26
=====] - 65s 3s/step - loss:
3.0693 - accuracy: 0.1347 - val_loss: 3.2082 -
val_accuracy: 0.0409\n\nEpoch 00037: val_loss did not
improve from 3.10871\n\nEpoch 00037: loss did not improve
from 3.04834\nEpoch 38/1000\n26/26
=====] - 65s 2s/step - loss:
3.0557 - accuracy: 0.1477 - val_loss: 3.1935 -
val_accuracy: 0.0409\n\nEpoch 00038: val_loss did not
improve from 3.10871\n\nEpoch 00038: loss did not improve
from 3.04834\nEpoch 39/1000\n26/26
=====] - 58s 2s/step - loss:
3.0518 - accuracy: 0.1562 - val_loss: 3.1678 -
val_accuracy: 0.0409\n\nEpoch 00039: val_loss did not
improve from 3.10871\n\nEpoch 00039: loss did not improve
from 3.04834\nEpoch 40/1000\n26/26
=====] - 56s 2s/step - loss:
3.0759 - accuracy: 0.1329 - val_loss: 3.1427 -
val_accuracy: 0.0409\n\nEpoch 00040: val_loss did not
improve from 3.10871\n\nEpoch 00040: loss did not improve
from 3.04834\nEpoch 41/1000\n26/26
=====] - 59s 2s/step - loss:

```



```

\001b[0
mclass_weight\001b[0m\001b[0;34m=\001b[0m\001b[0
mclass_weight\001b[0m\001b[0;34m,\001b[0m\001b[0;34m\
u001b[0m\001b[0m\n\001b[0;32m--> 220\001b[0;31m
reset_metrics
=False)\n\001b[0m\001b[1;32m 221\001b[0m \001b[
[0;34m\001b[0m\001b[0m\n\001b[1;32m 222\001b[0m
\001b[0mouts\001b[0m \001b[0;34m=\
u001b[0m \001b[0mto_list\001b[0m\001b[0;34m(\001b[0m\
u001b[0mouts\001b[0m\001b[0;34m)\001b[0m\001b[0;34m\
u001b[0m\001b[0m\n" ,
"\001b[0;32m/Library/Frameworks/Python.framework/Versions
/2.7/lib/python2.7/site-packages/keras/engine/training.
pyc\001b[0m in \001b[0;36mtrain_on_batch\001b[0;34m(
self, x, y, sample_weight, class_weight, reset_metrics)\
u001b[0m\n\001b[1;32m 1512\001b[0m \001b[
[0mins\001b[0m \001b[0;34m=\001b[0m \001b[0mx\001b[0
m \001b[0;34m+\001b[0m \001b[0my\001b[0m \001b[0;34m
+\001b[0m \001b[0msample_weights\001b[0m\001b[0;34m\
u001b[0m\001b[0m\n\001b[1;32m 1513\001b[0m \
\001b[0mself\001b[0m\001b[0;34m.\001b[0m\001b[0
m_make_train_function\001b[0m\001b[0;34m(\001b[0m\
u001b[0;34m)\001b[0m\001b[0;34m\001b[0m\001b[0m\n\
u001b[0;32m-> 1514\001b[0;31m \001b[0moutputs\
u001b[0m \001b[0;34m=\001b[0m \001b[0mself\001b[0m\
u001b[0;34m.\001b[0m\001b[0mtrain_function\001b[0m\
u001b[0;34m(\001b[0m\001b[0mins\001b[0m\001b[0;34m)\
u001b[0m\001b[0;34m\001b[0m\001b[0m\n\001b[0m\001b[
1;32m 1515\001b[0m \001b[0;34m\001b[0m\001b[0m\n\
u001b[1;32m 1516\001b[0m \001b[0;32mif\001b[
0m \001b[0mreset_metrics\001b[0m\001b[0;34m:\001b[0m\
u001b[0;34m\001b[0m\001b[0m\n" ,
"\001b[0;32m/Library/Frameworks/Python.framework/Versions
/2.7/lib/python2.7/site-packages/tensorflow_core/python/
keras/backend.pyc\001b[0m in \001b[0;36m__call__\001b[
0;34m(self, inputs)\001b[0m\n\001b[1;32m 3725\001b[
0m \001b[0mvalue\001b[0m \001b[0;34m=\001b[0
m \001b[0mmath_ops\001b[0m\001b[0;34m.\001b[0m\001b[
0mcast\001b[0m\001b[0;34m(\001b[0m\001b[0mvalue\
u001b[0m\001b[0;34m,\001b[0m \001b[0mtensor\001b[0m\
u001b[0;34m.\001b[0m\001b[0mdtype\001b[0m\001b[0;34m)
\001b[0m\001b[0;34m\001b[0m\n\001b[1;32m
3726\001b[0m \001b[0mconverted_inputs\001b[0m\
u001b[0;34m.\001b[0m\001b[0mappend\001b[0m\001b[0;34m(
\001b[0m\001b[0mvalue\001b[0m\001b[0;34m)\001b[0m\

```



```

m_call_flat\001b[0m\001b[0;34m(\001b[0m\001b[0margs\
001b[0m\001b[0;34m,\001b[0m \001b[0mself\001b[0m\
001b[0;34m.\001b[0m\001b[0m\001b[0mcaptured_inputs\001b[0m\
001b[0;34m,\001b[0m \001b[0m\001b[0mcancellation_manager\001b
[0m\001b[0;34m)\001b[0m\001b[0;34m\001b[0m\001b[0m\n
\001b[0m\001b[1;32m 1592\001b[0m \001b[0;34m\001b
[0m\001b[0m\n\001b[1;32m 1593\001b[0m \001b[0;32
mdef\001b[0m \001b[0m_filtered_call\001b[0m\001b[0;34
m(\001b[0m\001b[0mself\001b[0m\001b[0;34m,\001b[0m \001b[0m \
001b[0margs\001b[0m\001b[0;34m,\001b[0m \001b[0m \001b[0
mkwargs\001b[0m\001b[0;34m)\001b[0m\001b[0;34m:\001b
[0m\001b[0;34m\001b[0m\001b[0m\n",
"\001b[0;32m/Library/Frameworks/Python.framework/Versions
/2.7/lib/python2.7/site-packages/tensorflow_core/python/
eager/function.pyc\001b[0m in \001b[0;36m_call_flat\
001b[0;34m(self, args, captured_inputs,
cancellation_manager)\001b[0m\n\001b[1;32m 1690\001b[0m
[0m \001b[0;31m# No tape is watching; skip to
running the function.\001b[0m\001b[0;34m\001b[0m\001b[0m\001b[0
[0;34m\001b[0m\001b[0m\n\001b[1;32m 1691\001b[0m
return self._build_call_outputs(self.
_inference_function.call(\n\001b[0;32m-> 1692\001b[0;31
m ctx, args, cancellation_manager=
cancellation_manager))\n\001b[0m\001b[1;32m 1693\
001b[0m forward_backward = self.
_select_forward_and_backward_functions(\n\001b[1;32m
1694\001b[0m \001b[0margs\001b[0m\001b[0;34m
,\001b[0m\001b[0;34m\001b[0m\001b[0m\n",
"\001b[0;32m/Library/Frameworks/Python.framework/Versions
/2.7/lib/python2.7/site-packages/tensorflow_core/python/
eager/function.pyc\001b[0m in \001b[0;36m_call\001b
[0;34m(self, ctx, args, cancellation_manager)\001b[0m\n\
001b[1;32m 543\001b[0m \001b[0minputs
\001b[0m\001b[0;34m=\001b[0m\001b[0margs\001b[0m\
001b[0;34m,\001b[0m\001b[0;34m\001b[0m\001b[0m\n\
001b[1;32m 544\001b[0m \001b[0mattrs\
001b[0m\001b[0;34m=\001b[0m\001b[0;34m(\001b[0m\
001b[0;34m"executor_type"\001b[0m\001b[0;34m,\001b[0m
[0m \001b[0mexecutor_type\001b[0m\001b[0;34m,\001b[0m
\001b[0;34m"config_proto"\001b[0m\001b[0;34m,\001b[0m
[0m \001b[0mconfig\001b[0m\001b[0;34m)\001b[0m\001b[0
[0;34m,\001b[0m\001b[0;34m\001b[0m\001b[0m\n\001b
[0;32m-> 545\001b[0;31m ctx=ctx)\n\001b
[0m\001b[1;32m 546\001b[0m \001b[0;32melse\
001b[0m\001b[0;34m:\001b[0m\001b[0;34m\001b[0m\001b[0m

```

```

        outputs =
        execute.execute_with_cancellation("\n",
"\u001b[0;32m/Library/Frameworks/Python.framework/Versions
/2.7/lib/python2.7/site-packages/tensorflow_core/python/
eager/execute.pyc\u001b[0m in \u001b[0;36mquick_execute\u
\u001b[0;34m(op_name, num_outputs, inputs, attrs, ctx,
name)\u001b[0m\n\u001b[1;32m      59\u001b[0m      tensors
= pywrap_tensorflow.TFE_Py_Execute(ctx._handle,
device_name,\n\u001b[1;32m      60\u001b[0m      \u001b[0
\u001b[0mop_name\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0m\u001b[0minputs\u
\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0m\u001b[0mattrs\u001b[0m\u001b[0m\u001b[0
\u001b[0;34m,\u001b[0m\u001b[0m\u001b[0;\u001b[0;34m\u001b[0m\u001b[0m\u001b[0m\n\u
\u001b[0;32m--> 61\u001b[0;31m

num_outputs)\n\u001b[0m\u001b[1;32m      62\u001b[0m      \
\u001b[0;32mexcept\u001b[0m \u001b[0m\u001b[0mcore\u001b[0m\u001b[0m\u001b[0
\u001b[0;34m.\u001b[0m\u001b[0m\u001b[0m_NotOkStatusException\u001b[0m \u001b[0
\u001b[0;32mas\u001b[0m \u001b[0m\u001b[0m\u001b[0m\u001b[0;\u001b[0;34m:\u001b[0
\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\u001b[0m\n\u001b[1;32m
63\u001b[0m      \u001b[0;32mif\u001b[0m \u001b[0m\u001b[0mname\u001b[0
\u001b[0;32mis\u001b[0m \u001b[0m\u001b[0;\u001b[0;32mnot\u001b[0m \u001b[0m
\u001b[0m\u001b[0mNone\u001b[0m\u001b[0;34m:\u001b[0m \u001b[0m\u001b[0;\u001b[0;34m
\u001b[0m\u001b[0m\u001b[0m\n",
"\u001b[0;31mKeyboardInterrupt\u001b[0m: "
]
}
],
"source": "hist = model.fit_generator(\n      train_generator
,\n      steps_per_epoch= len(train_generator.filesnames)
//64,\n      epochs=1000,\n      shuffle = True,\n
      validation_data=test_generator,\n
validation_steps= 2,\n      callbacks=checkpoint)"
},
{
"cell_type": "code",
"execution_count": null,
"metadata": {},
"outputs": [],
"source": ""
}
],
"metadata": {
"kernel_spec": {
"display_name": "Python 2",

```

```
    "language": "python",
    "name": "python2"
  },
  "language_info": {
    "codemirror_mode": {
      "name": "ipython",
      "version": 2
    },
    "file_extension": ".py",
    "mimetype": "text/x-python",
    "name": "python",
    "nbconvert_exporter": "python",
    "pygments_lexer": "ipython2",
    "version": "2.7.16"
  }
},
"nbformat": 4,
"nbformat_minor": 4
}
```
