

```
1 import datetime
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from keras.layers import Input, Activation, Conv2D, Flatten, Dense, MaxPooling2D
5 from keras.models import Model, load_model
6 from keras.preprocessing.image import ImageDataGenerator
7 from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau
8 plt.style.use('dark_background')
```

Using TensorFlow backend.

## Load Dataset

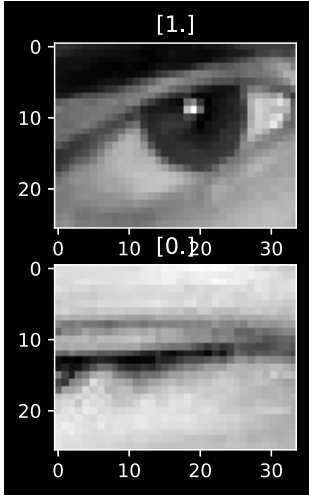
```
1 x_train = np.load('dataset/x_train.npy').astype(np.float32)
2 y_train = np.load('dataset/y_train.npy').astype(np.float32)
3 x_val = np.load('dataset/x_val.npy').astype(np.float32)
4 y_val = np.load('dataset/y_val.npy').astype(np.float32)
5
6 print(x_train.shape, y_train.shape)
7 print(x_val.shape, y_val.shape)
```

(2586, 26, 34, 1) (2586, 1)  
(288, 26, 34, 1) (288, 1)

## Preview

```
1 plt.subplot(2, 1, 1)
2 plt.title(str(y_train[0]))
3 plt.imshow(x_train[0].reshape((26, 34)), cmap='gray')
4 plt.subplot(2, 1, 2)
5 plt.title(str(y_val[4]))
6 plt.imshow(x_val[4].reshape((26, 34)), cmap='gray')
```

<matplotlib.image.AxesImage at 0x1a302c7af98>



## Data Augmentation

```
1 train_datagen = ImageDataGenerator(
2     rescale=1./255,
3     rotation_range=10,
4     width_shift_range=0.2,
5     height_shift_range=0.2,
6     shear_range=0.2
7 )
8
9 val_datagen = ImageDataGenerator(rescale=1./255)
10
11 train_generator = train_datagen.flow(
12     x=x_train, y=y_train,
13     batch_size=32,
14     shuffle=True
15 )
16
17 val_generator = val_datagen.flow(
18     x=x_val, y=y_val,
19     batch_size=32,
20     shuffle=False
21 )
```

## Build Model

```
1 inputs = Input(shape=(26, 34, 1))
2
3 net = Conv2D(32, kernel_size=3, strides=1, padding='same', activation='relu')(inputs)
4 net = MaxPooling2D(pool_size=2)(net)
5
6 net = Conv2D(64, kernel_size=3, strides=1, padding='same', activation='relu')(net)
7 net = MaxPooling2D(pool_size=2)(net)
8
9 net = Conv2D(128, kernel_size=3, strides=1, padding='same', activation='relu')(net)
10 net = MaxPooling2D(pool_size=2)(net)
11
12 net = Flatten()(net)
13
14 net = Dense(512)(net)
15 net = Activation('relu')(net)
16 net = Dense(1)(net)
17 outputs = Activation('sigmoid')(net)
18
19 model = Model(inputs=inputs, outputs=outputs)
20
21 model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['acc'])
22
23 model.summary()
```

WARNING:tensorflow:From C:\WPython36\lib\site-packages\keras\backend\tensorflow\_backend.py:74: The name tf.get\_default\_graph is deprecated. Please use tf.compat.v1.get\_default\_graph instead.

WARNING:tensorflow:From C:\Python36\lib\site-packages\tensorflow\tensorflow\_backend.py:517: The name tf.placeholder is deprecated. Please use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From C:\Python36\lib\site-packages\tensorflow\tensorflow\_backend.py:3976: The name tf.nn.max\_pool is deprecated. Please use tf.nn.max\_pool2d instead.

WARNING:tensorflow:From C:\Python36\lib\site-packages\tensorflow\tensorflow\_optimizers.py:790: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

WARNING:tensorflow:From C:\Python36\lib\site-packages\tensorflow\tensorflow\_backend.py:3376: The name tf.log is deprecated. Please use tf.math.log instead.

WARNING:tensorflow:From C:\Python36\lib\site-packages\tensorflow\tensorflow\_core\tensorflow\_ops\tensorflow\_ops\_impl.py:183: where (from tensorflow.python.ops.array\_ops) is deprecated and will be removed in a future version. Instructions for updating:  
Use tf.where in 2.0, which has the same broadcast rule as np.where

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 26, 34, 1)	0
conv2d_1 (Conv2D)	(None, 26, 34, 32)	320
max_pooling2d_1 (MaxPooling2	(None, 13, 17, 32)	0
conv2d_2 (Conv2D)	(None, 13, 17, 64)	18496
max_pooling2d_2 (MaxPooling2	(None, 6, 8, 64)	0
conv2d_3 (Conv2D)	(None, 6, 8, 128)	73856
max_pooling2d_3 (MaxPooling2	(None, 3, 4, 128)	0
flatten_1 (Flatten)	(None, 1536)	0
dense_1 (Dense)	(None, 512)	786944
activation_1 (Activation)	(None, 512)	0
dense_2 (Dense)	(None, 1)	513
activation_2 (Activation)	(None, 1)	0
Total params: 880,129		
Trainable params: 880,129		
Non-trainable params: 0		

## Train

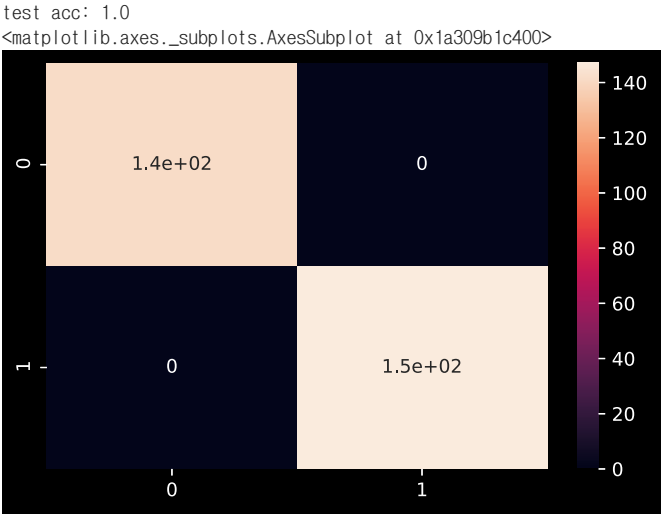
```
1 start_time = datetime.datetime.now().strftime('%Y_%m_%d_%H_%M_%S')
2
3 model.fit_generator(
4     train_generator, epochs=50, validation_data=val_generator,
5     steps_per_epoch = len(train_generator), validation_steps = len(val_generator),
6     callbacks=[
7         ModelCheckpoint('models/%s.h5' % (start_time), monitor='val_acc', save_best_only=True, mode='max', verbose=1),
8         ReduceLROnPlateau(monitor='val_acc', factor=0.2, patience=10, verbose=1, mode='auto', min_lr=1e-05)
9     ]
10 )
```

Epoch 37/50  
81/81 [=====] - 2s 28ms/step - loss: 0.0064 - acc: 0.9973 - val\_loss: 0.0027 - val\_acc: 1.0000  
  
Epoch 00037: val\_acc did not improve from 1.00000  
Epoch 38/50  
81/81 [=====] - 2s 27ms/step - loss: 0.0065 - acc: 0.9985 - val\_loss: 0.0025 - val\_acc: 1.0000  
  
Epoch 00038: val\_acc did not improve from 1.00000  
Epoch 39/50  
81/81 [=====] - 2s 27ms/step - loss: 0.0058 - acc: 0.9980 - val\_loss: 0.0076 - val\_acc: 0.9965  
  
Epoch 00039: val\_acc did not improve from 1.00000  
Epoch 40/50  
81/81 [=====] - 2s 27ms/step - loss: 0.0074 - acc: 0.9965 - val\_loss: 0.0033 - val\_acc: 1.0000  
  
Epoch 00040: val\_acc did not improve from 1.00000  
Epoch 41/50  
81/81 [=====] - 2s 27ms/step - loss: 0.0045 - acc: 0.9988 - val\_loss: 0.0068 - val\_acc: 0.9965  
  
Epoch 00041: val\_acc did not improve from 1.00000  
Epoch 42/50  
81/81 [=====] - 2s 26ms/step - loss: 0.0057 - acc: 0.9981 - val\_loss: 0.0010 - val\_acc: 1.0000  
  
Epoch 00042: val\_acc did not improve from 1.00000  
Epoch 43/50  
81/81 [=====] - 2s 26ms/step - loss: 0.0082 - acc: 0.9977 - val\_loss: 0.0013 - val\_acc: 1.0000  
  
Epoch 00043: val\_acc did not improve from 1.00000  
Epoch 44/50  
81/81 [=====] - 2s 27ms/step - loss: 0.0074 - acc: 0.9977 - val\_loss: 0.0063 - val\_acc: 0.9965  
  
Epoch 00044: val\_acc did not improve from 1.00000  
  
Epoch 00044: ReduceLROnPlateau reducing learning rate to 4.0000001899898055e-05.  
Epoch 45/50  
81/81 [=====] - 2s 28ms/step - loss: 0.0075 - acc: 0.9969 - val\_loss: 0.0071 - val\_acc: 0.9965  
  
Epoch 00045: val\_acc did not improve from 1.00000  
Epoch 46/50  
81/81 [=====] - 2s 28ms/step - loss: 0.0036 - acc: 0.9985 - val\_loss: 0.0057 - val\_acc: 0.9965  
  
Epoch 00046: val\_acc did not improve from 1.00000  
Epoch 47/50  
81/81 [=====] - 2s 28ms/step - loss: 0.0043 - acc: 0.9985 - val\_loss: 0.0061 - val\_acc: 0.9965  
  
Epoch 00047: val\_acc did not improve from 1.00000  
Epoch 48/50  
81/81 [=====] - 2s 27ms/step - loss: 0.0028 - acc: 0.9996 - val\_loss: 0.0044 - val\_acc: 0.9965  
  
Epoch 00048: val\_acc did not improve from 1.00000  
Epoch 49/50  
81/81 [=====] - 2s 27ms/step - loss: 0.0033 - acc: 0.9988 - val\_loss: 0.0045 - val\_acc: 0.9965  
  
Epoch 00049: val\_acc did not improve from 1.00000  
Epoch 50/50  
81/81 [=====] - 2s 27ms/step - loss: 0.0037 - acc: 0.9996 - val\_loss: 0.0027 - val\_acc: 1.0000  
  
Epoch 00050: val\_acc did not improve from 1.00000  
<keras.callbacks.History at 0x1a305f260b8>

## Confusion Matrix

```
1 from sklearn.metrics import accuracy_score, confusion_matrix
2 import seaborn as sns
3
4 model = load_model('models/%s.h5' % (start_time))
```

```
6 y_pred = model.predict(x_val/255.)
7 y_pred_logical = (y_pred > 0.5).astype(np.int)
8
9 print ('test acc: %s' % accuracy_score(y_val, y_pred_logical))
10 cm = confusion_matrix(y_val, y_pred_logical)
11 sns.heatmap(cm, annot=True)
```



▼ Distribution of Prediction

```
1 ax = sns.distplot(y_pred, kde=False)
```

C:\Python36\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version  
warnings.warn(msg, FutureWarning)

