In [1]:

```
import numpy as np
import pandas as pd

import os
print(os.listdir("./data/input"))

from keras import models

os.environ["CUDA_DEVICE_ORDER"]="PCI_BUS_ID"
os.environ["CUDA_VISIBLE_DEVICES"]="1"
```

```
['sample_submission.csv', 'train.csv', 'test_images', 'test.csv', 'train_images']
```

```
Using TensorFlow backend.
```

Load train data:

In [2]:

```
train_df = pd.read_csv('./data/input/train.csv')
test_df = pd.read_csv('./data/input/test.csv')
```

In [3]:

```
print(train_df.shape)
print(test_df.shape)
train_df.head(5)
```

```
(3662, 2)
(1928, 1)
```

Out[3]:

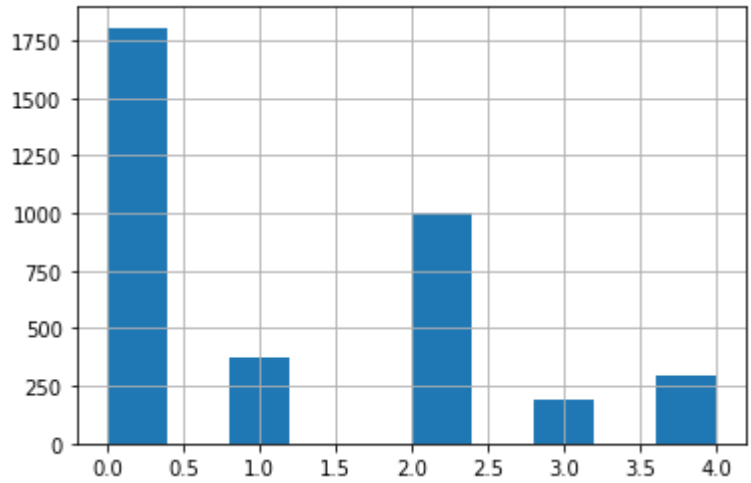
	id_code	diagnosis
0	000c1434d8d7	2
1	001639a390f0	4
2	0024cdab0c1e	1
3	002c21358ce6	0
4	005b95c28852	0

In [6]:

```
train_df['diagnosis'].value_counts()
train_df['diagnosis'].hist()
```

Out[6]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f56ce0dbbe0>
```



In [7]:

```
train_df['diagnosis'] = train_df['diagnosis'].astype('str')
train_df['id_code'] = train_df['id_code'].astype(str)+'.png'
```

In [8]:

```
train_df['id_code'][0]
```

Out[8]:

```
'000c1434d8d7.png'
```

In [9]:

```
import imageio as im
```

In [13]:

```
img = im.imread('./data/input/train_images/ff8a0b45c789.png')
print(img.shape)
```

```
(1226, 1844, 3)
```

In [14]:

```
train_df['diagnosis'][0]
```

Out[14]:

```
'2'
```

In [15]:

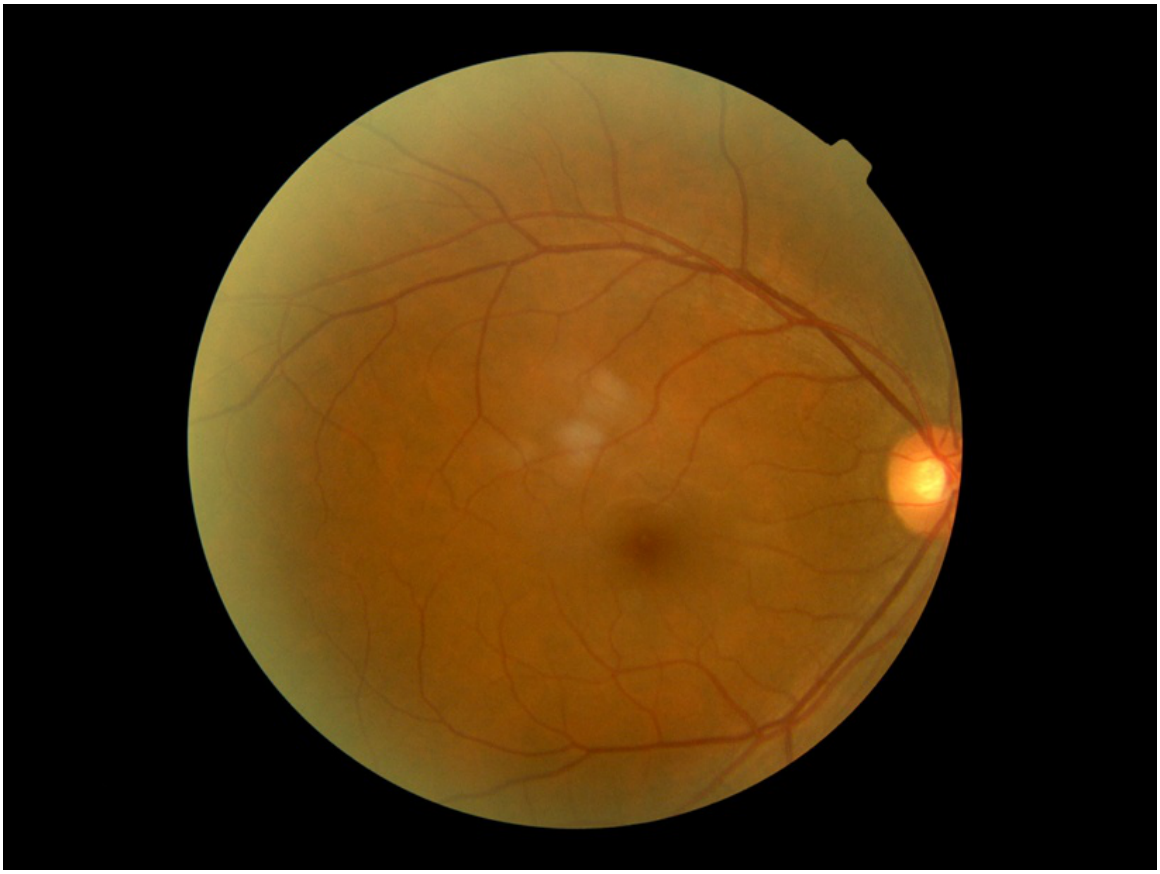
```
from IPython.display import Image
```

In [12]:

```
idx = 77
print(train_df['diagnosis'][idx])
Image('data/input/train_images/{}'.format(train_df['id_code'][idx]))
```

0

Out[12]:



In [13]:

```
import matplotlib.pyplot as plt
%matplotlib inline
import matplotlib.image as mpimg
import seaborn as sns
from keras.preprocessing import image
```

In [14]:

```
idx = 2
print(train_df['diagnosis'][idx])
img_path = 'data/input/train_images/{}'.format(train_df['id_code'][idx])
img = image.load_img(img_path, target_size=(96, 96))
img_tensor = image.img_to_array(img)
img_tensor = np.expand_dims(img_tensor, axis=0)
img_tensor /= 255.
plt.imshow(img_tensor[0])
plt.axis("off")
plt.show()
print(img_tensor.shape)
```

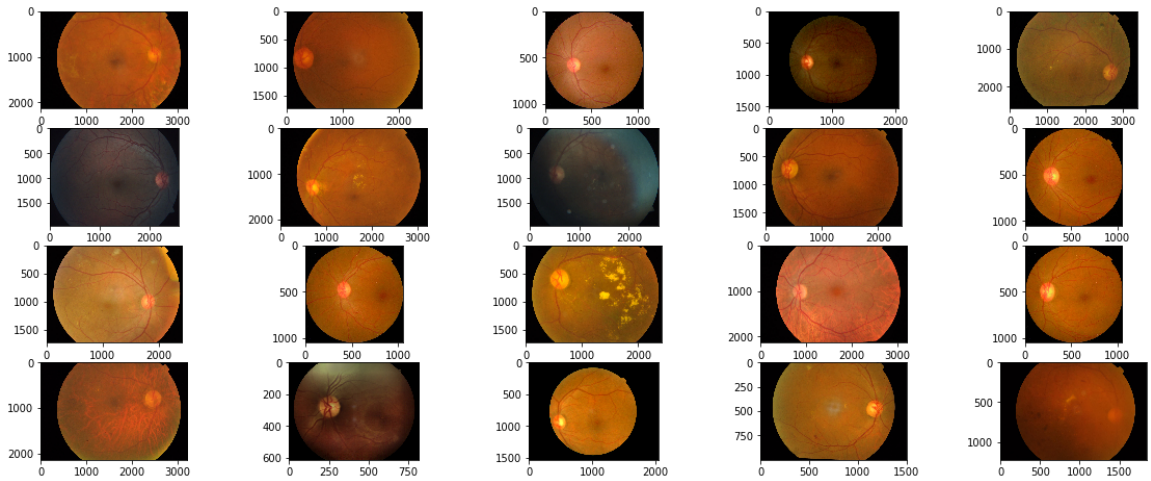
1

(1, 96, 96, 3)

In [15]:

```
import glob
images = []
for idx in range (1,21,1):
    print(idx, '-->', train_df['diagnosis'][idx])
    for img_path in glob.glob('data/input/train_images/{}'.format(train_df['id_code'][idx])):
        images.append(mpimg.imread(img_path))
plt.figure(figsize=(20,10))
columns = 5
idxx = 0
for i, image in enumerate(images):
    plt.subplot(len(images) / columns + 1, columns, i + 1)
    plt.imshow(image)
    #idxx += 1
    #plt.title('[{}]->{}'.format(i+1, train_df['diagnosis'][idxx]))
    #plt.show()
```

```
1 --> 4
2 --> 1
3 --> 0
4 --> 0
5 --> 4
6 --> 0
7 --> 2
8 --> 2
9 --> 1
10 --> 0
11 --> 2
12 --> 0
13 --> 3
14 --> 1
15 --> 0
16 --> 2
17 --> 0
18 --> 0
19 --> 2
20 --> 2
```



Function

- to get image from respective directory(train_images, test_images)
- to resize the large image

In [16]:

```
from keras.preprocessing.image import ImageDataGenerator

datagen=ImageDataGenerator(
    rescale=1./255,
    validation_split=0.2)

batch_size = 32

train_gen=datagen.flow_from_dataframe(
    dataframe=train_df,
    directory="./data/input/train_images",
    x_col="id_code",
    y_col="diagnosis",
    batch_size=batch_size,
    shuffle=True,
    class_mode="categorical",
    target_size=(96,96),
    subset='training')

test_gen=datagen.flow_from_dataframe(
    dataframe=train_df,
    directory="./data/input/train_images",
    x_col="id_code",
    y_col="diagnosis",
    batch_size=batch_size,
    shuffle=True,
    class_mode="categorical",
    target_size=(96,96),
    subset='validation')
```

```
Found 2930 validated image filenames belonging to 5 classes.
Found 732 validated image filenames belonging to 5 classes.
```

- Extract target column from training data
- Convert target column to categorical

In [17]:

```
y_train = train_df['diagnosis']
print(y_train)
from keras.utils import np_utils
y_train = np_utils.to_categorical(y_train)
print(y_train)
num_classes = y_train.shape[1]
print(num_classes)
```

0	2
1	4
2	1
3	0
4	0
5	4
6	0
7	2
8	2
9	1
10	0
11	2
12	0
13	3
14	1
15	0
16	2
17	0
18	0
19	2
20	2
21	0
22	1
23	2
24	0
25	2
26	0
27	0
28	0
29	0
	. .
3632	0
3633	0
3634	3
3635	1
3636	2
3637	0
3638	0
3639	1
3640	1
3641	0
3642	1
3643	0
3644	0
3645	0
3646	2
3647	0
3648	2
3649	0
3650	0
3651	0
3652	2
3653	0
3654	0
3655	2
3656	4
3657	2
3658	0
3659	2
3660	0
3661	2

```
Name: diagnosis, Length: 3662, dtype: object
[[0. 0. 1. 0. 0.]
 [0. 0. 0. 0. 1.]
 [0. 1. 0. 0. 0.]
 ...
 [0. 0. 1. 0. 0.]
 [1. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0.]]
5
```

Traditional CNN:

In [18]:

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout, GaussianNoise, GaussianDropout
from keras.layers import Flatten, BatchNormalization
from keras.layers.convolutional import Conv2D, SeparableConv2D
from keras.constraints import maxnorm
from keras.layers.convolutional import MaxPooling2D
from keras.utils import np_utils
from keras import backend as K
from keras import regularizers, optimizers
```

In [19]:

```
def build_model():
    # create model
    model = Sequential()
    #model.add(Reshape((x_train.shape[0],),))
    #model.add(GaussianDropout(0.3,input_shape=[96,96,3]))
    model.add(Conv2D(15, (3, 3), input_shape=[96,96,3], padding = 'Same', activation='relu'))
    model.add(GaussianDropout(0.3))
    model.add(Conv2D(30, (5, 5), activation='relu', padding = 'Same', kernel_constraint=maxnorm(
3)))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Conv2D(30, (3, 3), activation='relu', padding = 'Same'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Conv2D(60, (5, 5), activation='relu', padding = 'Same'))
    model.add(Conv2D(60, (7, 7), activation='relu', padding = 'Same'))

    model.add(Dropout(0.2))
    model.add(Flatten())
    model.add(Dense(256, activation='relu', kernel_regularizer=regularizers.l2(0.01)))
    model.add(Dense(128, activation='relu'))
    model.add(Dense(128, activation='relu'))
    model.add(Dense(50, activation='relu'))
    model.add(Dense(num_classes, activation='softmax', kernel_regularizer=regularizers.l2(0.0001
)
                    ,activity_regularizer=regularizers.l1(0.01)))
    # Compile model
    model.compile(loss='categorical_crossentropy', optimizer=optimizers.adam(lr=0.0001, amsgrad=
True), metrics=['accuracy'])
    return model
```

In [20]:

```
model = build_model()
```

```
WARNING: Logging before flag parsing goes to stderr.
W0718 11:45:57.468614 140216183326528 deprecation_wrapper.py:119] From /home/hyejo
o/.venv/py36tf/lib/python3.6/site-packages/keras/backend/tensorflow_backend.py:74:
The name tf.get_default_graph is deprecated. Please use tf.compat.v1.get_default_g
raph instead.

W0718 11:45:57.482578 140216183326528 deprecation_wrapper.py:119] From /home/hyejo
o/.venv/py36tf/lib/python3.6/site-packages/keras/backend/tensorflow_backend.py:51
7: The name tf.placeholder is deprecated. Please use tf.compat.v1.placeholder inst
ead.

W0718 11:45:57.483909 140216183326528 deprecation_wrapper.py:119] From /home/hyejo
o/.venv/py36tf/lib/python3.6/site-packages/keras/backend/tensorflow_backend.py:413
8: The name tf.random_uniform is deprecated. Please use tf.random.uniform instead.

W0718 11:45:57.492040 140216183326528 deprecation_wrapper.py:119] From /home/hyejo
o/.venv/py36tf/lib/python3.6/site-packages/keras/backend/tensorflow_backend.py:13
3: The name tf.placeholder_with_default is deprecated. Please use tf.compat.v1.pla
ceholder_with_default instead.

W0718 11:45:57.495911 140216183326528 deprecation_wrapper.py:119] From /home/hyejo
o/.venv/py36tf/lib/python3.6/site-packages/keras/backend/tensorflow_backend.py:411
5: The name tf.random_normal is deprecated. Please use tf.random.normal instead.

W0718 11:45:57.507862 140216183326528 deprecation_wrapper.py:119] From /home/hyejo
o/.venv/py36tf/lib/python3.6/site-packages/keras/backend/tensorflow_backend.py:397
6: The name tf.nn.max_pool is deprecated. Please use tf.nn.max_pool2d instead.

W0718 11:45:57.534038 140216183326528 deprecation.py:506] From /home/hyejoo/.venv/
py36tf/lib/python3.6/site-packages/keras/backend/tensorflow_backend.py:3445: calli
ng dropout (from tensorflow.python.ops.nn_ops) with keep_prob is deprecated and wi
ll be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_p
rob`.
W0718 11:45:57.595854 140216183326528 deprecation_wrapper.py:119] From /home/hyejo
o/.venv/py36tf/lib/python3.6/site-packages/keras/optimizers.py:790: The name tf.tr
ain.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.
```

In [21]:

```
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 96, 96, 15)        420
_________________________________________________________________
gaussian_dropout_1 (Gaussian (None, 96, 96, 15)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 96, 96, 30)        11280
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 48, 48, 30)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 48, 48, 30)        8130
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 24, 24, 30)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 24, 24, 60)        45060
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 24, 24, 60)        176460
_________________________________________________________________
dropout_1 (Dropout)          (None, 24, 24, 60)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 34560)             0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               8847616
_________________________________________________________________
dense_2 (Dense)              (None, 128)               32896
_________________________________________________________________
dense_3 (Dense)              (None, 128)               16512
_________________________________________________________________
dense_4 (Dense)              (None, 50)                6450
_________________________________________________________________
dense_5 (Dense)              (None, 5)                 255
=================================================================
Total params: 9,145,079
Trainable params: 9,145,079
Non-trainable params: 0
_________________________________________________________________
```

In [22]:

```
3*3*3*15+15
```

Out[22]:

420

In [23]:

```
5*5*15*30+30
```

Out[23]:

11280

In [24]:

```
3*3*30*30+30
```

Out[24]:

8130

In [25]:

```
5*5*30*50+50
```

Out[25]:

37550

In [26]:

```
7*7*50*50+50
```

Out[26]:

122550

In [27]:

```
6050*256+256
```

Out[27]:

1549056

In [28]:

```
256*128+128
```

Out[28]:

32896

In [29]:

```
128*50+50
```

Out[29]:

6450

In [30]:

```
50*5+5
```

Out[30]:

255

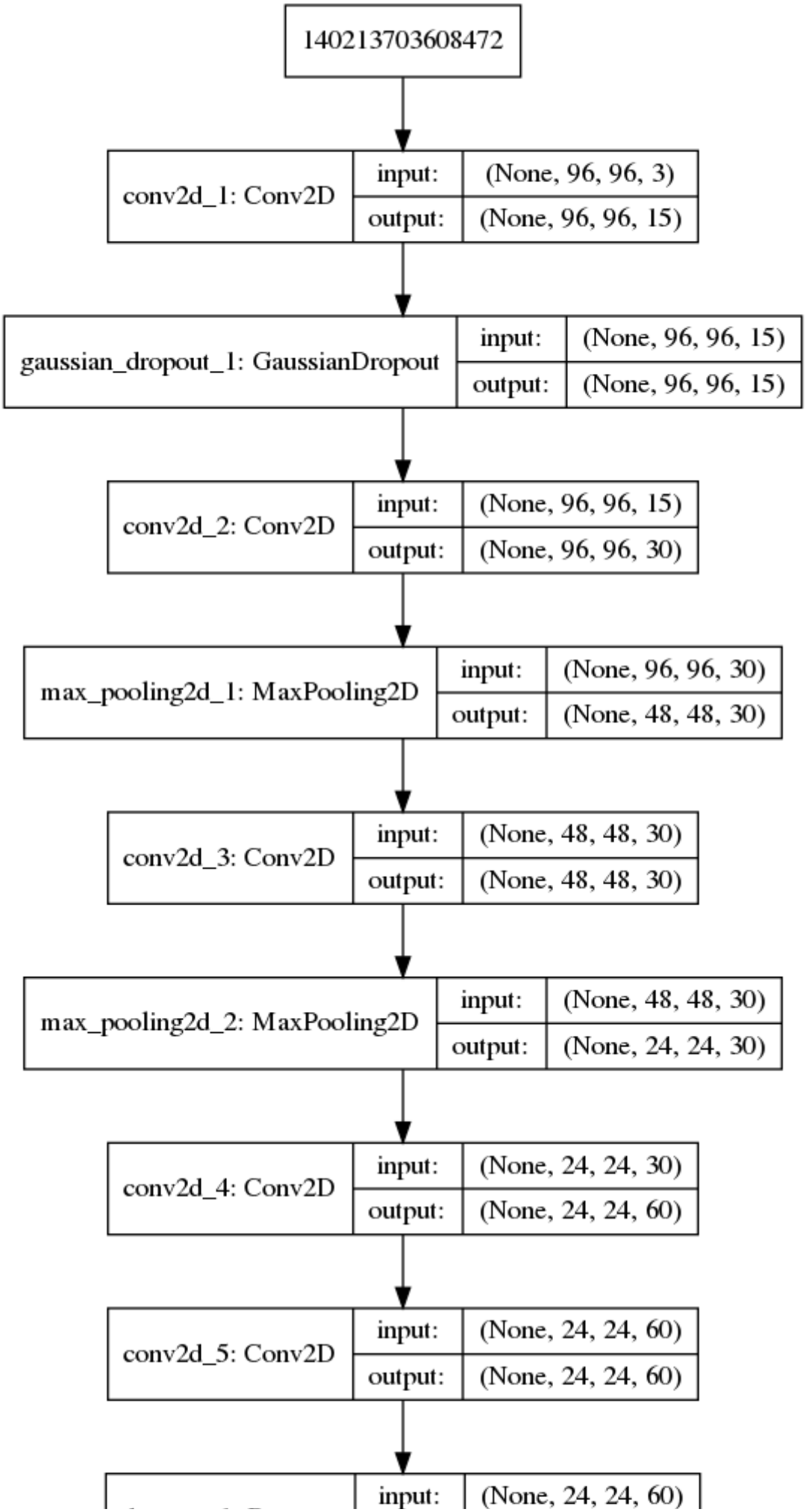
To prevent overfitting,

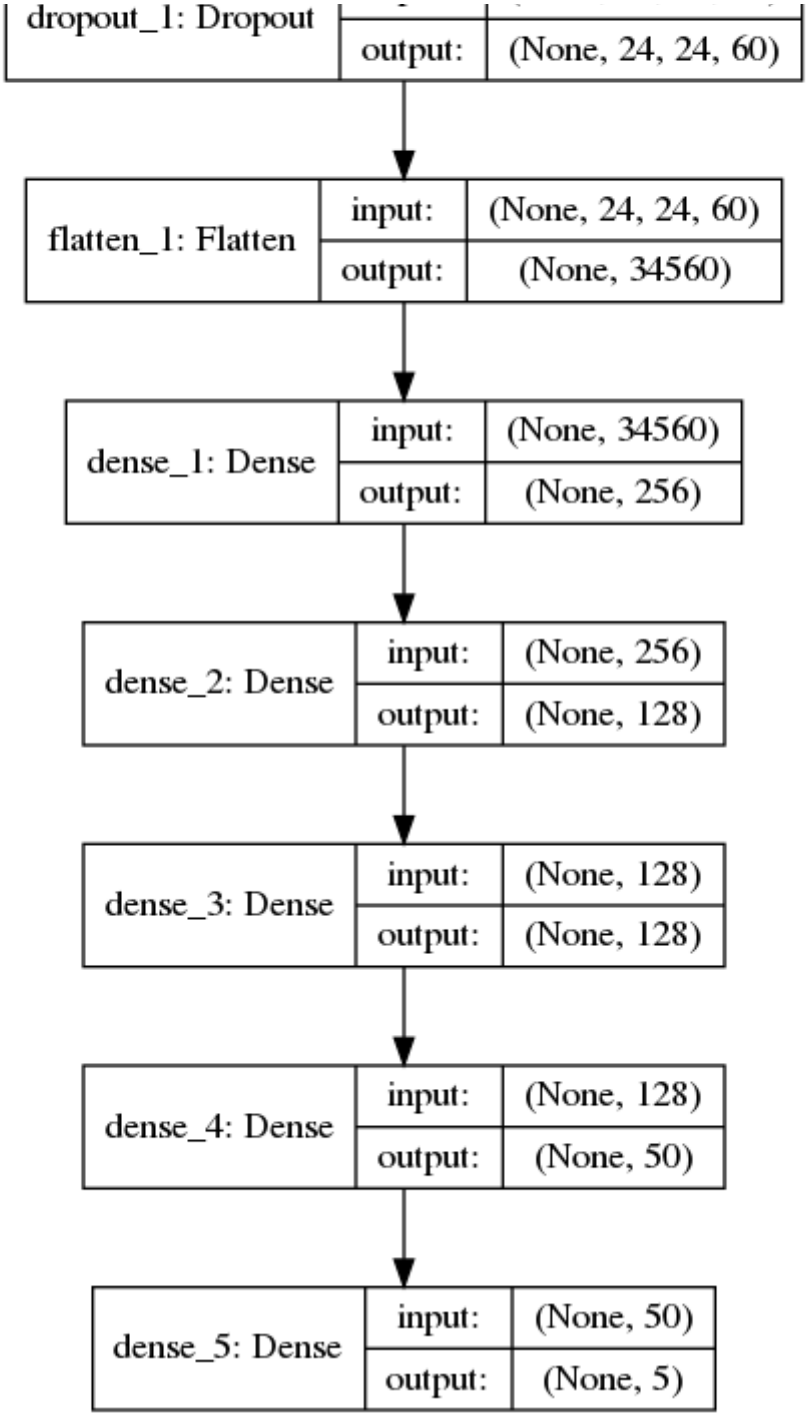
- monitoring the loss on validation/test set for minimum value
- run epochs for 20 times when there is no decrease in val_loss
- save the best model that has low validation loss

In [31]:

```
from keras.utils import plot_model

plot_model(model, to_file='./model.png', show_shapes=True)
plot_model(model, to_file='./model.svg', show_shapes=True)
from IPython.display import Image
Image('./model.png')
```

Out[31]:


140213703608472
conv2d_1: Conv2D
input: (None, 96, 96, 3)
output: (None, 96, 96, 15)
gaussian_dropout_1: GaussianDropout
input: (None, 96, 96, 15)
output: (None, 96, 96, 15)
conv2d_2: Conv2D
input: (None, 96, 96, 15)
output: (None, 96, 96, 30)
max_pooling2d_1: MaxPooling2D
input: (None, 96, 96, 30)
output: (None, 48, 48, 30)
conv2d_3: Conv2D
input: (None, 48, 48, 30)
output: (None, 48, 48, 30)
max_pooling2d_2: MaxPooling2D
input: (None, 48, 48, 30)
output: (None, 24, 24, 30)
conv2d_4: Conv2D
input: (None, 24, 24, 30)
output: (None, 24, 24, 60)
conv2d_5: Conv2D
input: (None, 24, 24, 60)
output: (None, 24, 24, 60)
input: (None, 24, 24, 60)


dropout_1: Dropout
output: (None, 24, 24, 60)
flatten_1: Flatten
input: (None, 24, 24, 60)
output: (None, 34560)
dense_1: Dense
input: (None, 34560)
output: (None, 256)
dense_2: Dense
input: (None, 256)
output: (None, 128)
dense_3: Dense
input: (None, 128)
output: (None, 128)
dense_4: Dense
input: (None, 128)
output: (None, 50)
dense_5: Dense
input: (None, 50)
output: (None, 5)

In [32]:

```
#from tensorflow.keras.callbacks import Callback
from keras.callbacks import Callback
from keras import backend as K

vloss = []
vacc = []
class NBatchLogger(Callback):

    def __init__(self, display):
        #self.step = 0
        self.display = display
        #self.metric_cache = {}

    #epoch 마다 learning rate 값 출력
    def on_epoch_end(self, epoch, logs=None):
        if self.display==1:
            print('aaaaa')
        global vloss
        global vacc

        vloss.append(logs['loss'])
        vacc.append(logs['acc'])
```

In [33]:

```
nbatch_logging = NBatchLogger(display=1)
```

In [34]:

```
from keras.callbacks import EarlyStopping, ModelCheckpoint
es= EarlyStopping(monitor='val_loss', mode ='min', verbose = 0, patience = 20)
mc = ModelCheckpoint('model.h5', monitor='val_loss', save_best_only = True, mode ='min', verbose
= 0)
```

In [35]:

```
history = model.fit_generator(generator=train_gen,
                                    steps_per_epoch=len(train_gen),
                                    validation_data=test_gen,
                                    validation_steps=len(test_gen),
                                    epochs=3,
                                    callbacks = [es, mc, nbatch_logging],
                                    use_multiprocessing = True,
                                    verbose=1)
```

```
W0718 11:45:58.113449 140216183326528 deprecation.py:323] From /home/hyejoo/.venv/
py36tf/lib/python3.6/site-packages/tensorflow/python/ops/math_grad.py:1250: add_di
spatch_support.<locals>.wrapper (from tensorflow.python.ops.array_ops) is deprecat
ed 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

Epoch 1/3
92/92 [==============================] - 227s 2s/step - loss: 3.8845 - acc: 0.5387
- val_loss: 2.3649 - val_acc: 0.6571
aaaaa
Epoch 2/3
92/92 [==============================] - 200s 2s/step - loss: 1.8209 - acc: 0.6955
- val_loss: 1.6612 - val_acc: 0.6940
aaaaa
Epoch 3/3
92/92 [==============================] - 198s 2s/step - loss: 1.5165 - acc: 0.7078
- val_loss: 1.5083 - val_acc: 0.7077
aaaaa
```

In [36]:

```
print("-- Evaluate --")
scores = model.evaluate_generator(train_gen, steps=5)
print("%s: %.2f%%" %(model.metrics_names[1], scores[1]*100))
```

```
-- Evaluate --
acc: 70.00%
```

In [37]:

```
print("-- Evaluate --")
scores = model.evaluate_generator(test_gen, steps=5)
print("%s: %.2f%%" %(model.metrics_names[1], scores[1]*100))
```

```
-- Evaluate --
acc: 72.50%
```

In [38]:

```
from keras.models import load_model
model = load_model('model.h5')
```

In [39]:

```
model.summary()
```

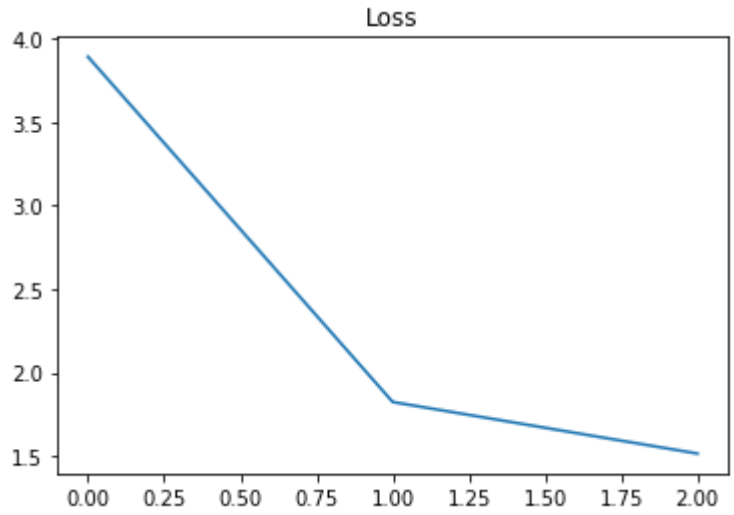
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 96, 96, 15)        420
_________________________________________________________________
gaussian_dropout_1 (Gaussian (None, 96, 96, 15)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 96, 96, 30)        11280
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 48, 48, 30)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 48, 48, 30)        8130
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 24, 24, 30)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 24, 24, 60)        45060
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 24, 24, 60)        176460
_________________________________________________________________
dropout_1 (Dropout)          (None, 24, 24, 60)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 34560)             0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               8847616
_________________________________________________________________
dense_2 (Dense)              (None, 128)               32896
_________________________________________________________________
dense_3 (Dense)              (None, 128)               16512
_________________________________________________________________
dense_4 (Dense)              (None, 50)                6450
_________________________________________________________________
dense_5 (Dense)              (None, 5)                 255
=================================================================
Total params: 9,145,079
Trainable params: 9,145,079
Non-trainable params: 0
_________________________________________________________________
```

In [40]:

```
plt.plot(vloss)
plt.title('Loss')
```

Out[40]:

```
Text(0.5, 1.0, 'Loss')
```

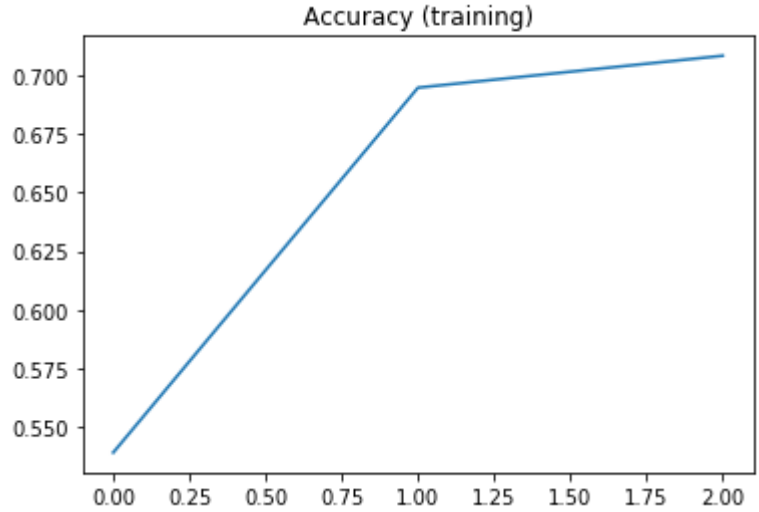


In [41]:

```
plt.plot(vacc)
plt.title('Accuracy (training)')
```

Out[41]:

```
Text(0.5, 1.0, 'Accuracy (training)')
```



In [2]:

```
acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(0, len(acc) + 1)
plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.legend()
plt.figure()
plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-2-01f2839ea65d> in <module>
----> 1 acc = history.history['acc']
      2 val_acc = history.history['val_acc']
      3 loss = history.history['loss']
      4 val_loss = history.history['val_loss']
      5 epochs = range(0, len(acc) + 1)

NameError: name 'history' is not defined
```

In [43]:

```
from sklearn.metrics import classification_report, confusion_matrix

#Confution Matrix and Classification Report
Y_pred = model.predict_generator(test_gen, 23)#//num_of_test_samples // batch_size+1
y_pred = np.argmax(Y_pred, axis=1)
```

In [44]:

```
print('Confusion Matrix')
A = confusion_matrix(test_gen.classes, y_pred)
print(A)
```

```
Confusion Matrix
[[166  60 111   0   0]
 [ 34  13  30   0   0]
 [111  38  72   0   0]
 [ 20   8  10   0   0]
 [ 23  14  22   0   0]]
```

In [45]:

```
print(y_pred.shape)
print(np.shape(test_gen.classes))
```

```
(732,)
(732,)
```

In [46]:

```
print(y_pred)
print(len(y_pred))
print(np.array(test_gen.classes))
```

```
[1 0 0 0 2 2 2 0 0 0 0 2 2 0 1 0 0 0 2 0 2 0 0 1 0 0 2 0 0 0 2 2 0 0 0 1 1
 1 0 2 2 2 2 0 2 1 2 0 0 0 0 2 0 2 0 2 0 2 0 2 2 0 2 2 2 2 1 2 1 2 2 2 0 0
 2 0 2 0 2 0 2 0 1 0 0 2 0 0 2 0 2 0 2 0 2 1 0 0 2 0 0 2 1 0 2 0 2 2 0 1 0 0 2
 2 0 0 0 0 0 0 0 0 0 1 2 1 2 0 0 0 0 0 0 1 0 0 0 0 2 1 1 0 0 0 0 2 0 0 2 0 2 0
 2 0 1 0 1 1 2 2 2 2 0 0 1 0 2 2 0 1 2 0 1 0 2 1 0 2 1 0 1 0 2 2 0 2 2 2 2 2
 0 0 1 0 0 1 0 2 0 2 2 1 2 2 0 2 0 0 0 0 2 0 2 1 2 2 1 2 0 1 2 2 0 2 1 2 1
 0 2 2 2 1 0 1 0 0 0 1 0 2 0 1 0 1 0 2 0 0 0 0 1 0 1 0 2 1 2 1 1 0 0 2 2 2
 0 0 0 1 0 0 2 1 1 1 0 0 1 0 0 0 0 2 1 2 0 2 2 0 0 0 0 0 2 0 0 2 0 0 2 0 0
 2 2 2 2 0 0 0 0 0 1 0 0 0 0 0 2 0 2 1 1 0 0 0 0 2 0 2 2 1 2 1 0 2 0 1 2 2
 1 0 2 0 0 0 1 2 0 0 1 1 2 0 0 1 0 2 1 1 1 0 1 1 1 0 2 2 0 2 2 0 0 1 0 0 2
 2 0 2 0 0 2 0 2 2 0 2 2 2 2 0 2 0 1 0 0 0 0 1 1 0 1 0 1 0 1 2 0 0 0 2 2 0
 0 2 0 0 0 2 2 2 2 0 0 0 2 2 2 2 2 0 2 0 2 2 2 0 2 0 0 0 2 2 1 2 0 0 0 0 0
 2 0 0 0 2 2 2 0 2 1 0 1 1 0 2 2 2 0 2 0 0 1 0 0 0 1 0 0 0 0 0 2 0 0 2 2 0
 0 2 0 0 1 2 0 2 0 1 0 2 2 1 2 0 2 0 2 2 2 1 0 0 0 0 0 0 1 0 0 2 1 2 0 2 0
 2 2 0 1 2 2 2 0 1 0 2 0 2 0 1 1 2 0 2 1 1 2 0 0 0 2 1 2 0 0 2 2 2 2 2 0 2
 0 2 0 0 0 1 1 0 0 0 1 0 0 0 2 2 1 0 0 0 2 0 2 0 0 2 0 2 0 2 0 0 0 0 0 0 2 1
 2 2 2 0 2 0 0 0 0 0 0 0 0 0 2 0 2 0 2 0 1 2 2 0 0 0 1 1 2 1 1 2 2 0 0 0 2 0
 1 2 2 1 2 1 1 2 0 0 0 0 2 2 0 0 1 0 0 1 2 2 0 0 2 0 2 2 0 0 0 0 1 2 0 2 1
 1 2 0 1 2 1 1 1 0 0 2 1 0 1 2 0 0 2 2 2 1 0 2 1 2 0 0 1 0 1 0 0 1 2 0 1 0
 0 0 0 0 0 0 2 1 2 0 1 1 0 1 0 0 1 0 0 0 0 0 1 2 0 2 1 0 1]
732
[2 4 1 0 0 4 0 2 2 1 0 2 0 3 1 0 2 0 0 2 2 0 1 2 0 2 0 0 0 0 0 0 4 2 4 2 0
 0 4 0 4 2 2 4 2 1 2 4 0 3 1 2 2 2 2 0 3 0 2 1 0 2 0 0 0 0 2 2 0 0 0 1 1 0
 3 0 2 0 2 0 1 3 1 0 2 0 0 3 4 2 2 0 0 0 0 1 0 3 1 0 2 2 0 0 0 4 4 0 0 1 2
 4 0 0 2 0 0 0 0 0 3 0 2 2 0 0 4 0 0 1 0 2 0 2 2 2 1 0 0 1 0 0 0 2 0 2 1 4
 0 2 0 2 0 0 0 0 0 0 4 2 0 0 2 2 2 3 2 0 3 0 0 0 0 4 0 2 0 1 0 0 0 2 0 0 0
 0 2 0 1 3 1 0 4 0 0 2 4 0 1 0 3 0 2 2 0 1 0 3 3 1 2 0 0 2 2 0 2 0 0 2 1 0
 0 1 0 0 0 2 2 0 1 0 4 1 1 2 4 2 2 2 0 2 0 2 0 2 2 0 0 0 0 1 1 2 0 2 0 2 1
 0 0 0 2 0 2 2 2 2 0 2 2 2 0 3 2 0 0 0 0 2 2 0 2 2 0 3 0 1 2 0 4 3 0 0 0 0
 3 2 0 4 0 0 0 4 3 4 2 1 0 2 0 0 0 0 0 2 1 0 2 2 2 0 0 0 0 4 0 2 0 0 2 1 0
 0 0 0 2 2 0 2 2 3 0 3 2 1 0 2 2 1 0 0 1 4 1 0 0 2 0 2 4 4 3 0 2 0 2 1 4 0
 0 2 3 0 0 2 1 2 4 1 2 0 1 0 2 4 2 2 2 2 0 3 0 0 0 0 0 0 2 0 0 1 2 0 0 0 0
 0 2 1 0 2 0 0 2 0 2 2 0 0 0 1 2 0 0 2 2 2 2 0 0 1 0 4 0 2 0 0 4 2 2 0 0 0
 2 0 2 0 2 0 0 0 4 0 2 1 4 0 0 2 2 2 2 0 2 4 0 0 0 4 2 0 2 0 4 4 1 0 0 2 4
 1 2 0 2 0 0 4 1 2 0 2 2 2 0 2 2 2 0 3 0 2 2 2 0 0 0 2 2 4 4 1 0 3 1 0 0 0
 2 0 2 0 4 0 1 0 3 0 4 2 1 2 0 2 4 4 0 2 4 3 0 0 2 3 2 0 0 0 2 0 2 2 2 0 1
 0 2 1 0 2 2 4 1 0 2 4 0 2 0 0 0 0 2 0 2 0 0 2 1 2 0 2 0 0 0 2 2 2 0 3 1 0
 1 4 0 2 0 1 2 0 0 2 2 0 2 2 0 1 0 2 2 2 0 2 0 0 3 0 0 2 2 3 0 4 3 2 2 2 0
 0 0 0 0 2 2 0 2 0 0 0 0 0 2 0 1 2 1 0 0 0 0 4 2 2 0 1 0 0 0 0 4 1 3 1 2 4
 0 0 0 2 2 4 0 2 4 0 2 0 2 1 0 0 2 1 1 0 2 4 2 0 0 2 2 0 3 0 0 2 1 4 3 0 2
 0 0 0 2 0 0 1 0 2 0 0 0 3 0 1 0 0 0 0 2 1 2 2 0 3 0 1 2 0]
```

Visualization

In [47]:

```
layer_outputs = [layer.output for layer in model.layers[:12]]
# Extracts the outputs of the top 12 layers
activation_model = models.Model(inputs=model.input, outputs=layer_outputs) # Creates a model tha
t will return these outputs, given the model input
```

In [48]:

```
# 그냥 레이어 몇개인지 출력해봄
for i in range (0,15,1):
    print(model.layers[i])
```

```
<keras.layers.convolutional.Conv2D object at 0x7f8597c89358>
<keras.layers.noise.GaussianDropout object at 0x7f8597c898d0>
<keras.layers.convolutional.Conv2D object at 0x7f8597c89940>
<keras.layers.pooling.MaxPooling2D object at 0x7f8597c89630>
<keras.layers.convolutional.Conv2D object at 0x7f8597c7ef60>
<keras.layers.pooling.MaxPooling2D object at 0x7f859974db70>
<keras.layers.convolutional.Conv2D object at 0x7f859974d710>
<keras.layers.convolutional.Conv2D object at 0x7f860c17f3c8>
<keras.layers.core.Dropout object at 0x7f8597c41ac8>
<keras.layers.core.Flatten object at 0x7f8597c41ba8>
<keras.layers.core.Dense object at 0x7f860e482fd0>
<keras.layers.core.Dense object at 0x7f860b07b198>
<keras.layers.core.Dense object at 0x7f8597c98da0>
<keras.layers.core.Dense object at 0x7f860bff09e8>
<keras.layers.core.Dense object at 0x7f85996fa518>
```

In [49]:

```
activations = activation_model.predict(img_tensor)
# Returns a list of five Numpy arrays: one array per layer activation
```

In [50]:

```
first_layer_activation = activations[0]
print(first_layer_activation.shape)
```

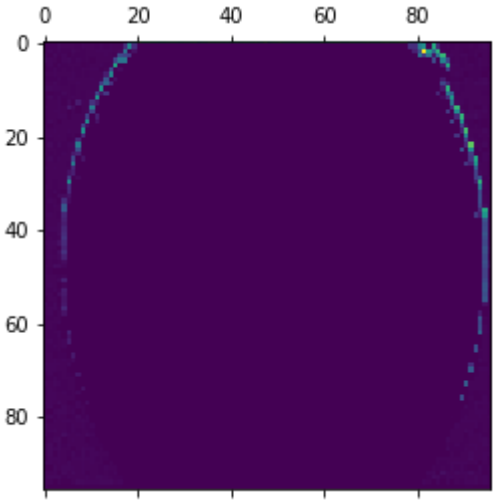
```
(1, 96, 96, 15)
```

In [51]:

```
plt.matshow(first_layer_activation[0, :, :, 4], cmap='viridis')
```

Out[51]:

```
<matplotlib.image.AxesImage at 0x7f85a5b10400>
```



In [52]:

```
print(np.shape(model.layers))
model.layers[:12]
```

(15,)

Out[52]:

```
[<keras.layers.convolutional.Conv2D at 0x7f8597c89358>,
 <keras.layers.noise.GaussianDropout at 0x7f8597c898d0>,
 <keras.layers.convolutional.Conv2D at 0x7f8597c89940>,
 <keras.layers.pooling.MaxPooling2D at 0x7f8597c89630>,
 <keras.layers.convolutional.Conv2D at 0x7f8597c7ef60>,
 <keras.layers.pooling.MaxPooling2D at 0x7f859974db70>,
 <keras.layers.convolutional.Conv2D at 0x7f859974d710>,
 <keras.layers.convolutional.Conv2D at 0x7f860c17f3c8>,
 <keras.layers.core.Dropout at 0x7f8597c41ac8>,
 <keras.layers.core.Flatten at 0x7f8597c41ba8>,
 <keras.layers.core.Dense at 0x7f860e482fd0>,
 <keras.layers.core.Dense at 0x7f860b07b198>]
```

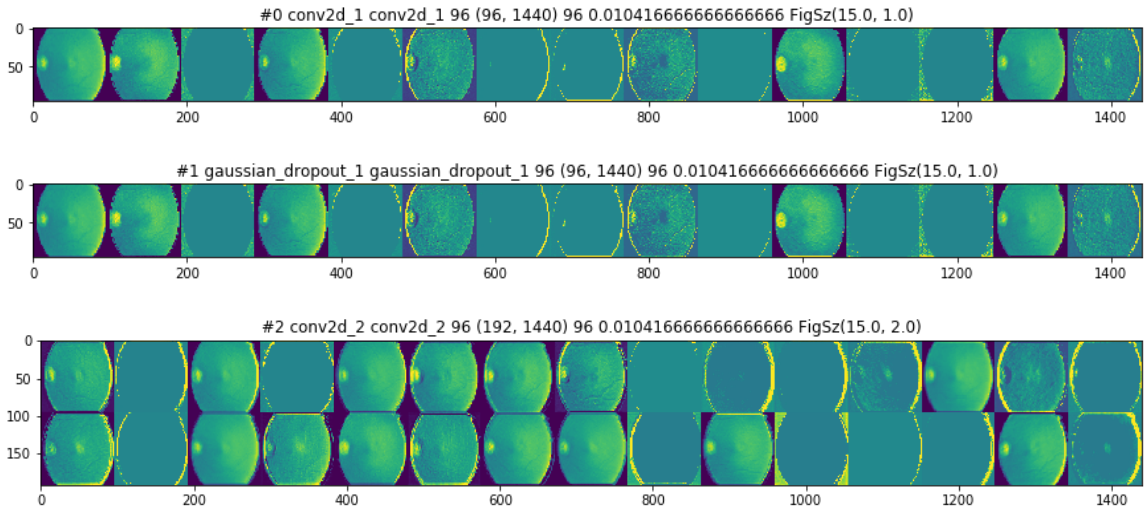
In [53]:

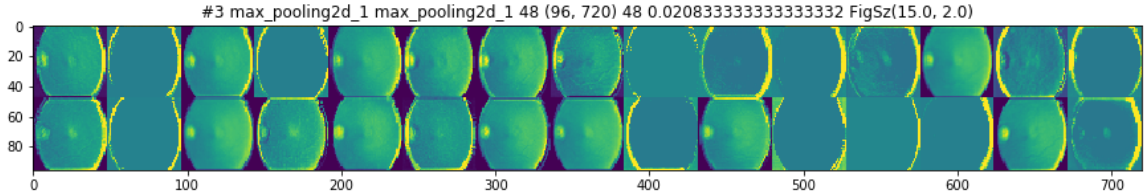
```
layer_names = []
for layer in model.layers[:12]:
    layer_names.append(layer.name) # Names of the layers, so you can have them as part of your p
lot

images_per_row = 15#6
```

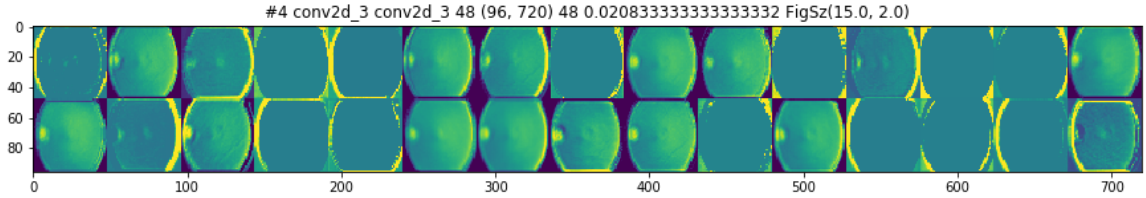
In [54]:

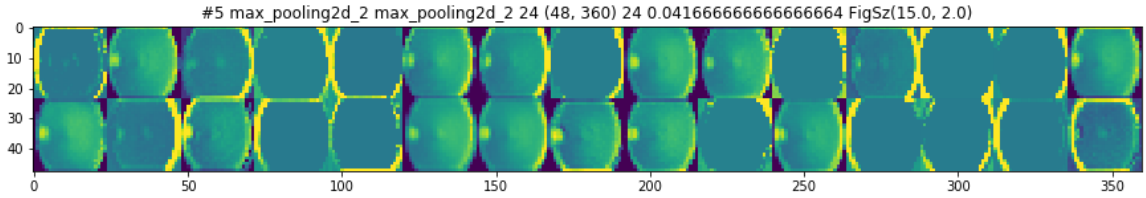
```
i=0
for layer_name, layer_activation in zip(layer_names, activations): # Displays the feature maps
    n_features = layer_activation.shape[-1] # Number of features in the feature map
    size = layer_activation.shape[1] #The feature map has shape (1, size, size, n_features).
    n_cols = n_features // images_per_row # Tiles the activation channels in this matrix
    display_grid = np.zeros((size * n_cols, images_per_row * size))
    for col in range(n_cols): # Tiles each filter into a big horizontal grid
        for row in range(images_per_row):
            channel_image = layer_activation[0,
                                             :, :,
                                             col * images_per_row + row]
            channel_image -= channel_image.mean() # Post-processes the feature to make it visual
ly palatable
            channel_image /= channel_image.std()
            channel_image *= 64
            channel_image += 128
            channel_image = np.clip(channel_image, 0, 255).astype('uint8')
            display_grid[col * size : (col + 1) * size, # Displays the grid
                         row * size : (row + 1) * size] = channel_image
    scale = 1. / size
    figsize=(scale * display_grid.shape[1], scale * display_grid.shape[0])
    print()
    title ='#{} {} {} {} {} {} {} FigSz{}'.format(i, layer_name, layer_name, size, display_grid
.shape, size, scale, figsize)
    print(title)
    plt.figure(figsize=figsize)
    plt.title(title)
    plt.grid(False)
    plt.imshow(display_grid, aspect='auto', cmap='viridis')

    #if i>3:
    #    break

    i=i+1
```

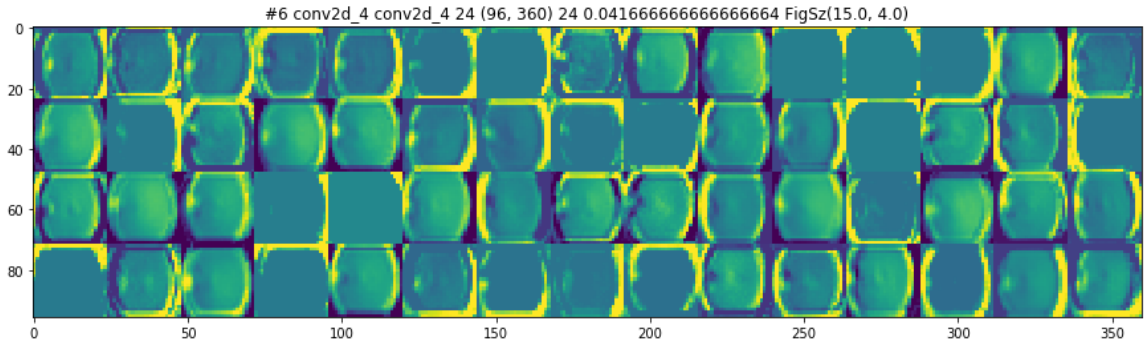
```
#0 conv2d_1 conv2d_1 96 (96, 1440) 96 0.010416666666666666 FigSz(15.0, 1.0)

#1 gaussian_dropout_1 gaussian_dropout_1 96 (96, 1440) 96 0.010416666666666666 Fig
Sz(15.0, 1.0)

#2 conv2d_2 conv2d_2 96 (192, 1440) 96 0.010416666666666666 FigSz(15.0, 2.0)

#3 max_pooling2d_1 max_pooling2d_1 48 (96, 720) 48 0.020833333333333332 FigSz(15.
0, 2.0)

#4 conv2d_3 conv2d_3 48 (96, 720) 48 0.020833333333333332 FigSz(15.0, 2.0)

#5 max_pooling2d_2 max_pooling2d_2 24 (48, 360) 24 0.041666666666666664 FigSz(15.
0, 2.0)

#6 conv2d_4 conv2d_4 24 (96, 360) 24 0.041666666666666664 FigSz(15.0, 4.0)

#7 conv2d_5 conv2d_5 24 (96, 360) 24 0.041666666666666664 FigSz(15.0, 4.0)

#8 dropout_1 dropout_1 24 (96, 360) 24 0.041666666666666664 FigSz(15.0, 4.0)

/home/hyejoo/.venv/py36tf/lib/python3.6/site-packages/ipykernel_launcher.py:13: Ru
ntimeWarning: invalid value encountered in true_divide
  del sys.path[0]

---------------------------------------------------------------------------
MemoryError                               Traceback (most recent call last)
<ipython-input-54-5a48849983e1> in <module>
      4     size = layer_activation.shape[1] #The feature map has shape (1, size,
 size, n_features).
      5     n_cols = n_features // images_per_row # Tiles the activation channels
 in this matrix
----> 6     display_grid = np.zeros((size * n_cols, images_per_row * size))
      7     for col in range(n_cols): # Tiles each filter into a big horizontal gr
id
      8         for row in range(images_per_row):

MemoryError:
```

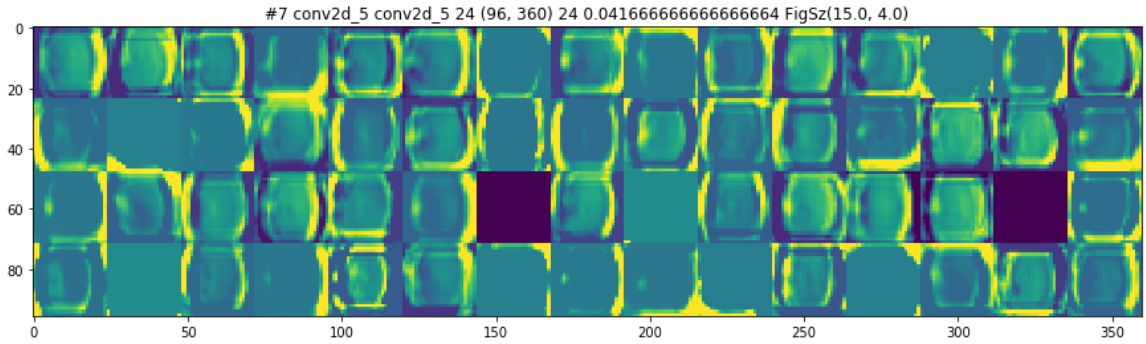


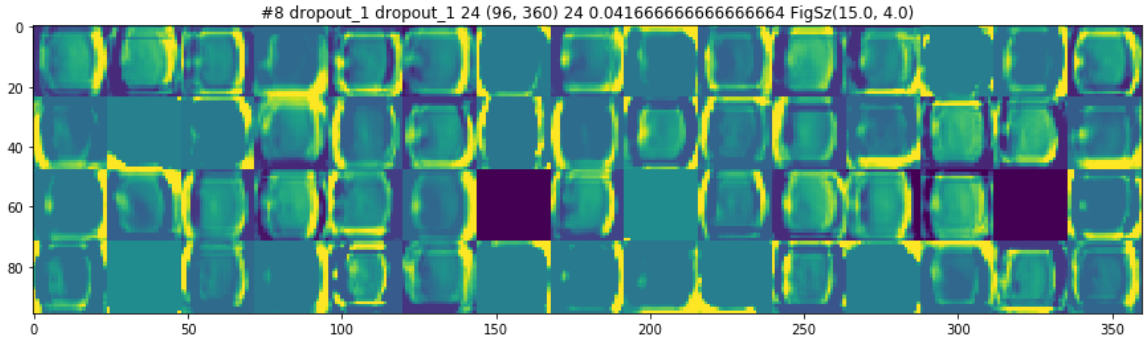

#3 max_pooling2d_1 max_pooling2d_1 48 (96, 720) 48 0.020833333333333332 FigSz(15.0, 2.0)


#4 conv2d_3 conv2d_3 48 (96, 720) 48 0.020833333333333332 FigSz(15.0, 2.0)


#5 max_pooling2d_2 max_pooling2d_2 24 (48, 360) 24 0.041666666666666664 FigSz(15.0, 2.0)


#6 conv2d_4 conv2d_4 24 (96, 360) 24 0.041666666666666664 FigSz(15.0, 4.0)


#7 conv2d_5 conv2d_5 24 (96, 360) 24 0.041666666666666664 FigSz(15.0, 4.0)


#8 dropout_1 dropout_1 24 (96, 360) 24 0.041666666666666664 FigSz(15.0, 4.0)

In [ ]:

In [ ]:

```
```