In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf

import numpy as np # linear algebra
import pandas as pd
from utils import *

import os
import numpy as np
from tempfile import TemporaryFile
import pandas as pd
import matplotlib.pyplot as plt
from tensorflow.keras.callbacks import TensorBoard, ModelCheckpoint
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Flatten, GaussianNoise
from tensorflow.keras.layers import Conv2D, MaxPooling2D
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.regularizers import l1
```

In [2]:
```
tf.enable_eager_execution()
```

This is the training pipeline:

-a classifier is trained for every chanel

-to use this pipepline change to value of the variable "idc" bellow and run each cell

In [3]:
```
idc = 0
```

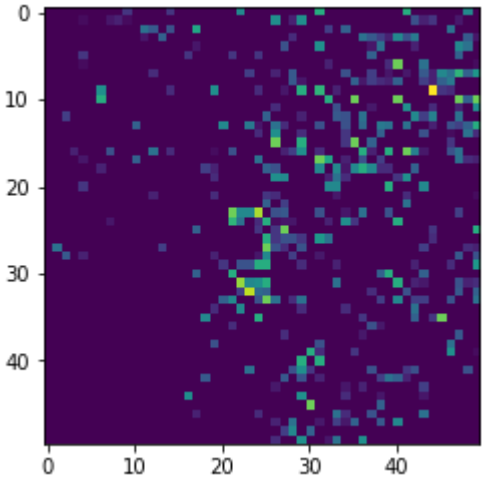
In [4]:
```
### Loads in the channel
xtest_chan = np.load("channels/channel_{}_xtest.npy".format(idc))
xtrain_chan = np.load("channels/channel_{}_xtrain.npy".format(idc))

### NB: Labels for each channel are the same
ytest_chan = np.load("channels/channel_5_ytest.npy")
ytrain_chan = np.load("channels/channel_5_ytrain.npy")
```

In [5]:
```
### obtain predictions for one-hot encoding
ytrain = np.argmax(ytrain_chan, axis = 1)
ytest = np.argmax(ytest_chan, axis = 1)
```

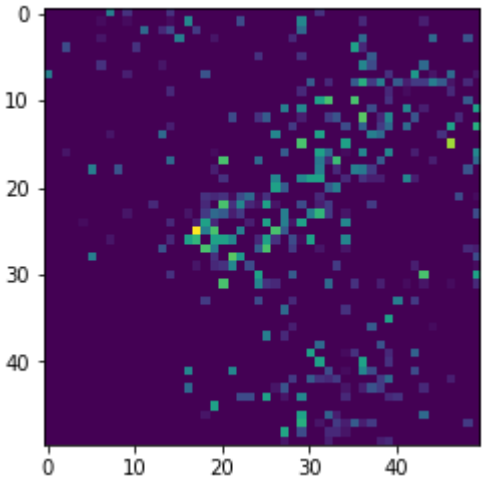
In [6]:
```
### Crop and resize the images
###  140*140 center crop was done and the image was then resiszed to 50*50
im_size = 50
xtrain = np.array(crop_resize(xtrain_chan, 140, im_size)).reshape((xtrain_chan
.shape[0], im_size,im_size, 1))
xtest = np.array(crop_resize(xtest_chan, 140, im_size)).reshape((xtest_chan.sh
ape[0], im_size,im_size, 1))
```

In [7]:
```
### Display the image and labels for 5 images in the dataset
i = 10 ###start index
for x in range(5):
    print(ytrain[x+i])
    plt.imshow(xtrain[x+i].reshape(im_size,im_size))
    plt.show()
```

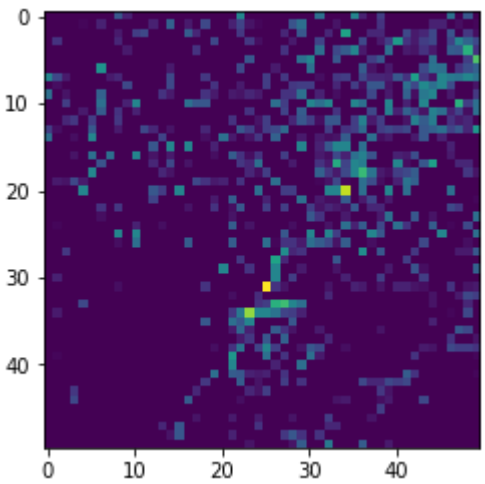
2


0
10
20
30
40
0
10
20
30
40

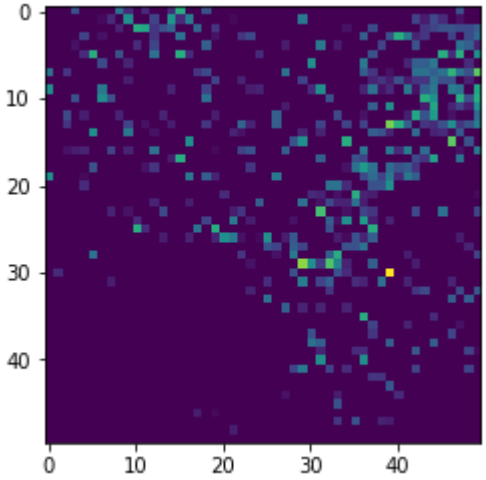
6


0
10
20
30
40
0
10
20
30
40

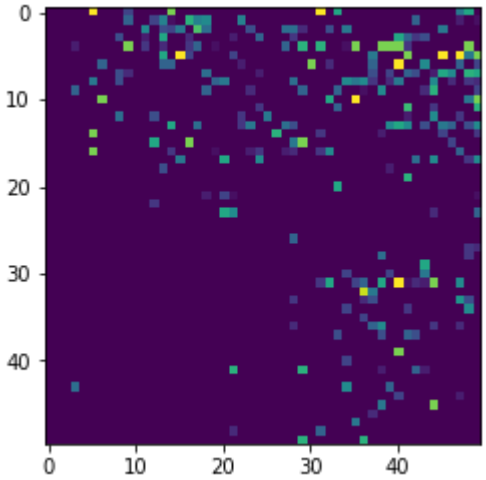
7


0
10
20
30
40
0
10
20
30
40

0

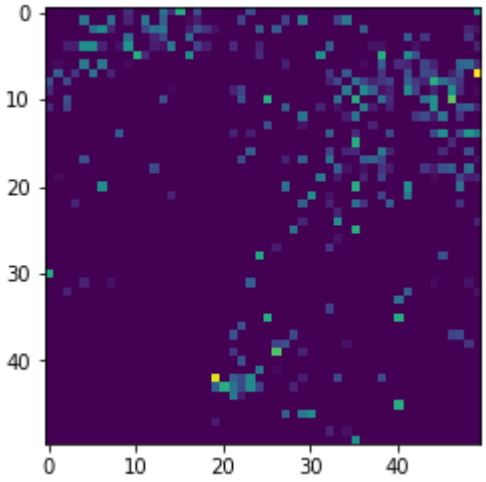


9

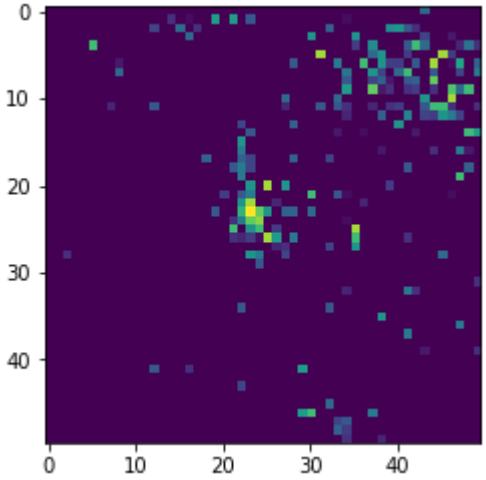


In [45]:
```
### Displays 5 images and their respective labels from the test set
i = 68
for x in range(5):
    print(ytest[x+i])
    plt.imshow(xtest[x+i].reshape(im_size,im_size))
    plt.show()
```

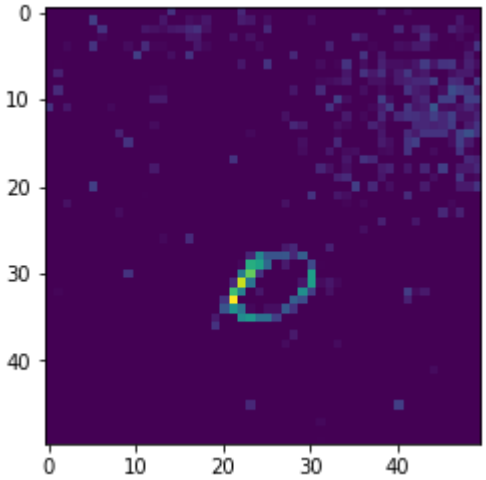
8


0
10
20
30
40
0
10
20
30
40

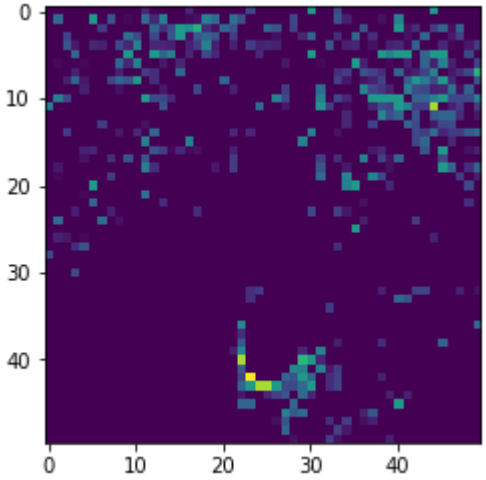
1


0
10
20
30
40
0
10
20
30
40

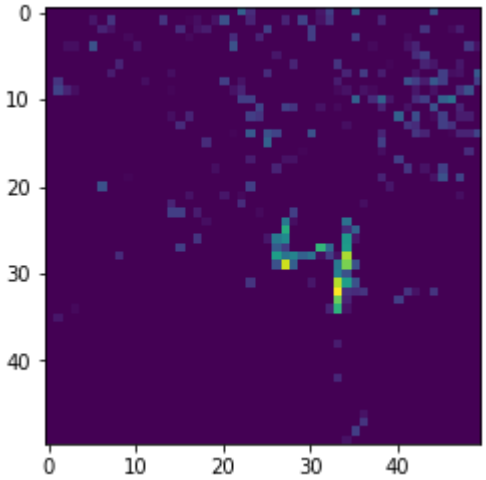
0


0
10
20
30
40
0
10
20
30
40

0



4



In [46]:
```
print(xtrain.shape)
print(ytrain.shape)
```

```
(477, 50, 50, 1)
(477,)
```

In [61]:
```
im_size =50
batch_size = 120
num_classes = 10
lr = 0.001 ### learning rate
input_shape = ( im_size, im_size, 1)
l1_lambda = 0.0003 ### L1 regularization lambda parameter
```

In [71]:
```
### Creating model architecture


he_init = tf.keras.initializers.VarianceScaling()
model = Sequential()
model.add(BatchNormalization(input_shape=input_shape))
model.add(Conv2D(64, (2, 2), kernel_regularizer=l1(l1_lambda), activation='el
u',
                 kernel_initializer =he_init))

model.add(GaussianNoise(0.01))
model.add(Conv2D(64, (2, 2), kernel_regularizer=l1(l1_lambda), activation='el
u',
                 kernel_initializer =he_init))

model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))
model.add(Conv2D(128, (2, 2), kernel_regularizer=l1(l1_lambda), activation='el
u',
                 kernel_initializer =he_init))

model.add(GaussianNoise(0.01))
model.add(Conv2D(128, (2, 2), kernel_regularizer=l1(l1_lambda), activation='el
u',
                 kernel_initializer =he_init))

model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))
model.add(Conv2D(256, (2, 2), kernel_regularizer=l1(l1_lambda), activation='el
u',
                 kernel_initializer =he_init))

model.add(GaussianNoise(0.1))
model.add(Conv2D(256, (2, 2), kernel_regularizer=l1(l1_lambda), activation='el
u',
                 kernel_initializer =he_init))

model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))

model.add(Flatten())
model.add(Dense(1024, activation='elu', kernel_regularizer=l1(l1_lambda), kern
el_initializer =he_init))
model.add(Dropout(0.2))

model.add(Dense(1024, activation='elu',kernel_initializer =he_init))
model.add(Dropout(0.2))
model.add(GaussianNoise(0.01))
model.add(Dense(num_classes, activation='softmax', kernel_initializer =he_init
))
model.summary()
```

Layer (type)	Output Shape	Param #
batch_normalization_v1_5 (Ba	(None, 50, 50, 1)	4
conv2d_30 (Conv2D)	(None, 49, 49, 64)	320
gaussian_noise_15 (GaussianN	(None, 49, 49, 64)	0
conv2d_31 (Conv2D)	(None, 48, 48, 64)	16448
max_pooling2d_15 (MaxPooling	(None, 24, 24, 64)	0
dropout_25 (Dropout)	(None, 24, 24, 64)	0
conv2d_32 (Conv2D)	(None, 23, 23, 128)	32896
gaussian_noise_16 (GaussianN	(None, 23, 23, 128)	0
conv2d_33 (Conv2D)	(None, 22, 22, 128)	65664
max_pooling2d_16 (MaxPooling	(None, 11, 11, 128)	0
dropout_26 (Dropout)	(None, 11, 11, 128)	0
conv2d_34 (Conv2D)	(None, 10, 10, 256)	131328
gaussian_noise_17 (GaussianN	(None, 10, 10, 256)	0
conv2d_35 (Conv2D)	(None, 9, 9, 256)	262400
max_pooling2d_17 (MaxPooling	(None, 4, 4, 256)	0
dropout_27 (Dropout)	(None, 4, 4, 256)	0
flatten_5 (Flatten)	(None, 4096)	0
dense_15 (Dense)	(None, 1024)	4195328
dropout_28 (Dropout)	(None, 1024)	0
dense_16 (Dense)	(None, 1024)	1049600
dropout_29 (Dropout)	(None, 1024)	0
gaussian_noise_18 (GaussianN	(None, 1024)	0
dense_17 (Dense)	(None, 10)	10250

Total params: 5,764,238
Trainable params: 5,764,236
Non-trainable params: 2

In [72]:
```
### This loads in saved weights only run if the notebook some how restart and
 you want to
### reinitialized trained weights
#model.load_weights("Models/model{}.h5".format(idc))
```

In [73]:
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

datagen = ImageDataGenerator(#rotation_range = 5,
                             width_shift_range = [1, 5],
                             height_shift_range = [1, 5],
                             #brightness_range = (.1, 5),
                             shear_range = 5.0
                            )
datagen.fit(xtrain)
```

In [74]:
```
path = "tf_keras_models/"
```

In [75]:
```
opt = tf.keras.optimizers.Adam(lr)
model.compile(loss='categorical_crossentropy',
              optimizer=opt,
              metrics=['accuracy'])

#tensorboard = TensorBoard(log_dir='./logs', write_graph=True)
#checkpoint = ModelCheckpoint(filepath=os.path.join(path, "model-{epoch:02d}.h
5"))
"""
model.fit_generator(datagen.flow(train_data, train_labels,
          batch_size=batch_size),
          epochs=epochs,
          verbose=1,
          validation_data=(test_data, test_labels),
          callbacks=[tensorboard])
"""
```

Out[75]:
```
'\nmodel.fit_generator(datagen.flow(train_data, train_labels,\n          batc
h_size=batch_size),\n          epochs=epochs,\n          verbose=1,\n
validation_data=(test_data, test_labels),\n          callbacks=[tensorboard])
\n'
```

In [76]:
```
epochs = 100
batch_size = 477
model.fit_generator(datagen.flow(xtrain, ytrain_chan, batch_size = batch_size
),
          epochs=epochs,
          verbose=1,
          validation_data=(xtest, ytest_chan),
          #callbacks=[ checkpointer]
         )
# Any results you write to the current directory are saved as
```

```
Epoch 1/100
236/236 [==============================] - 2s 7ms/sample - loss: 27.4217 - ac
c: 0.3008
1/1 [==============================] - 12s 12s/step - loss: 23.6170 - acc: 0.
8302 - val_loss: 27.4217 - val_acc: 0.3008
Epoch 2/100
236/236 [==============================] - 1s 3ms/sample - loss: 26.0198 - ac
c: 0.2458
1/1 [==============================] - 4s 4s/step - loss: 25.7552 - acc: 0.34
38 - val_loss: 26.0198 - val_acc: 0.2458
Epoch 3/100
236/236 [==============================] - 1s 3ms/sample - loss: 25.0508 - ac
c: 0.3347
1/1 [==============================] - 4s 4s/step - loss: 25.5207 - acc: 0.29
56 - val_loss: 25.0508 - val_acc: 0.3347
Epoch 4/100
236/236 [==============================] - 1s 3ms/sample - loss: 24.1238 - ac
c: 0.3686
1/1 [==============================] - 4s 4s/step - loss: 24.4032 - acc: 0.38
36 - val_loss: 24.1238 - val_acc: 0.3686
Epoch 5/100
236/236 [==============================] - 1s 3ms/sample - loss: 23.1236 - ac
c: 0.3686
1/1 [==============================] - 4s 4s/step - loss: 23.8070 - acc: 0.38
99 - val_loss: 23.1236 - val_acc: 0.3686
Epoch 6/100
236/236 [==============================] - 1s 3ms/sample - loss: 22.0129 - ac
c: 0.4661
1/1 [==============================] - 4s 4s/step - loss: 22.5961 - acc: 0.44
86 - val_loss: 22.0129 - val_acc: 0.4661
Epoch 7/100
236/236 [==============================] - 1s 3ms/sample - loss: 21.3849 - ac
c: 0.4958
1/1 [==============================] - 4s 4s/step - loss: 21.4432 - acc: 0.53
25 - val_loss: 21.3849 - val_acc: 0.4958
Epoch 8/100
236/236 [==============================] - 1s 3ms/sample - loss: 20.7141 - ac
c: 0.5254
1/1 [==============================] - 4s 4s/step - loss: 20.7289 - acc: 0.57
02 - val_loss: 20.7141 - val_acc: 0.5254
Epoch 9/100
236/236 [==============================] - 0s 2ms/sample - loss: 20.3765 - ac
c: 0.4576
1/1 [==============================] - 4s 4s/step - loss: 20.0927 - acc: 0.59
54 - val_loss: 20.3765 - val_acc: 0.4576
Epoch 10/100
236/236 [==============================] - 1s 3ms/sample - loss: 19.9994 - ac
c: 0.4703
1/1 [==============================] - 4s 4s/step - loss: 19.6910 - acc: 0.58
07 - val_loss: 19.9994 - val_acc: 0.4703
Epoch 11/100
236/236 [==============================] - 1s 3ms/sample - loss: 20.0370 - ac
c: 0.5042
1/1 [==============================] - 4s 4s/step - loss: 19.2652 - acc: 0.61
01 - val_loss: 20.0370 - val_acc: 0.5042
Epoch 12/100
236/236 [==============================] - 1s 2ms/sample - loss: 19.1770 - ac
```

```
c: 0.5339
1/1 [==============================] - 4s 4s/step - loss: 18.9574 - acc: 0.61
84 - val_loss: 19.1770 - val_acc: 0.5339
Epoch 13/100
236/236 [==============================] - 1s 3ms/sample - loss: 19.0163 - ac
c: 0.4958
1/1 [==============================] - 4s 4s/step - loss: 18.1766 - acc: 0.69
60 - val_loss: 19.0163 - val_acc: 0.4958
Epoch 14/100
236/236 [==============================] - 1s 3ms/sample - loss: 18.3620 - ac
c: 0.5636
1/1 [==============================] - 4s 4s/step - loss: 18.0940 - acc: 0.60
80 - val_loss: 18.3620 - val_acc: 0.5636
Epoch 15/100
236/236 [==============================] - 1s 3ms/sample - loss: 17.8803 - ac
c: 0.5466
1/1 [==============================] - 5s 5s/step - loss: 17.5543 - acc: 0.66
25 - val_loss: 17.8803 - val_acc: 0.5466
Epoch 16/100
236/236 [==============================] - 1s 3ms/sample - loss: 17.5605 - ac
c: 0.5127
1/1 [==============================] - 4s 4s/step - loss: 17.0933 - acc: 0.70
86 - val_loss: 17.5605 - val_acc: 0.5127
Epoch 17/100
236/236 [==============================] - 1s 3ms/sample - loss: 17.2718 - ac
c: 0.4958
1/1 [==============================] - 4s 4s/step - loss: 16.7280 - acc: 0.67
30 - val_loss: 17.2718 - val_acc: 0.4958
Epoch 18/100
236/236 [==============================] - 0s 2ms/sample - loss: 16.8437 - ac
c: 0.5212
1/1 [==============================] - 4s 4s/step - loss: 16.4055 - acc: 0.64
78 - val_loss: 16.8437 - val_acc: 0.5212
Epoch 19/100
236/236 [==============================] - 1s 3ms/sample - loss: 16.3762 - ac
c: 0.5466
1/1 [==============================] - 4s 4s/step - loss: 15.9739 - acc: 0.70
23 - val_loss: 16.3762 - val_acc: 0.5466
Epoch 20/100
236/236 [==============================] - 1s 3ms/sample - loss: 16.0858 - ac
c: 0.5212
1/1 [==============================] - 4s 4s/step - loss: 15.5631 - acc: 0.72
96 - val_loss: 16.0858 - val_acc: 0.5212
Epoch 21/100
236/236 [==============================] - 1s 3ms/sample - loss: 15.8890 - ac
c: 0.5212
1/1 [==============================] - 4s 4s/step - loss: 15.1510 - acc: 0.76
10 - val_loss: 15.8890 - val_acc: 0.5212
Epoch 22/100
236/236 [==============================] - 1s 3ms/sample - loss: 15.5157 - ac
c: 0.5551
1/1 [==============================] - 4s 4s/step - loss: 14.8474 - acc: 0.71
70 - val_loss: 15.5157 - val_acc: 0.5551
Epoch 23/100
236/236 [==============================] - 1s 3ms/sample - loss: 15.2092 - ac
c: 0.5720
1/1 [==============================] - 4s 4s/step - loss: 14.5192 - acc: 0.73
```

```
79 - val_loss: 15.2092 - val_acc: 0.5720
Epoch 24/100
236/236 [==============================] - 1s 3ms/sample - loss: 14.8783 - ac
c: 0.5720
1/1 [==============================] - 4s 4s/step - loss: 14.2168 - acc: 0.71
28 - val_loss: 14.8783 - val_acc: 0.5720
Epoch 25/100
236/236 [==============================] - 1s 2ms/sample - loss: 14.6348 - ac
c: 0.5381
1/1 [==============================] - 4s 4s/step - loss: 13.8165 - acc: 0.76
73 - val_loss: 14.6348 - val_acc: 0.5381
Epoch 26/100
236/236 [==============================] - 1s 3ms/sample - loss: 14.2326 - ac
c: 0.5720
1/1 [==============================] - 5s 5s/step - loss: 13.6354 - acc: 0.73
58 - val_loss: 14.2326 - val_acc: 0.5720
Epoch 27/100
236/236 [==============================] - 1s 3ms/sample - loss: 13.9678 - ac
c: 0.5466
1/1 [==============================] - 4s 4s/step - loss: 13.2666 - acc: 0.75
89 - val_loss: 13.9678 - val_acc: 0.5466
Epoch 28/100
236/236 [==============================] - 1s 3ms/sample - loss: 13.6598 - ac
c: 0.5508
1/1 [==============================] - 4s 4s/step - loss: 12.9755 - acc: 0.75
47 - val_loss: 13.6598 - val_acc: 0.5508
Epoch 29/100
236/236 [==============================] - 1s 2ms/sample - loss: 13.4319 - ac
c: 0.5297
1/1 [==============================] - 4s 4s/step - loss: 12.6311 - acc: 0.80
50 - val_loss: 13.4319 - val_acc: 0.5297
Epoch 30/100
236/236 [==============================] - 1s 3ms/sample - loss: 13.1376 - ac
c: 0.5551
1/1 [==============================] - 5s 5s/step - loss: 12.4395 - acc: 0.77
99 - val_loss: 13.1376 - val_acc: 0.5551
Epoch 31/100
236/236 [==============================] - 1s 2ms/sample - loss: 12.8185 - ac
c: 0.5593
1/1 [==============================] - 4s 4s/step - loss: 12.1964 - acc: 0.77
57 - val_loss: 12.8185 - val_acc: 0.5593
Epoch 32/100
236/236 [==============================] - 1s 3ms/sample - loss: 12.6065 - ac
c: 0.5424
1/1 [==============================] - 4s 4s/step - loss: 11.9041 - acc: 0.77
99 - val_loss: 12.6065 - val_acc: 0.5424
Epoch 33/100
236/236 [==============================] - 1s 2ms/sample - loss: 12.3878 - ac
c: 0.5339
1/1 [==============================] - 4s 4s/step - loss: 11.7097 - acc: 0.78
83 - val_loss: 12.3878 - val_acc: 0.5339
Epoch 34/100
236/236 [==============================] - 1s 3ms/sample - loss: 12.1870 - ac
c: 0.5636
1/1 [==============================] - 5s 5s/step - loss: 11.4717 - acc: 0.78
62 - val_loss: 12.1870 - val_acc: 0.5636
Epoch 35/100
```

```
236/236 [==============================] - 1s 2ms/sample - loss: 11.9571 - ac
c: 0.5720
1/1 [==============================] - 4s 4s/step - loss: 11.2141 - acc: 0.80
08 - val_loss: 11.9571 - val_acc: 0.5720
Epoch 36/100
236/236 [==============================] - 1s 3ms/sample - loss: 11.6927 - ac
c: 0.5508
1/1 [==============================] - 4s 4s/step - loss: 11.0170 - acc: 0.81
55 - val_loss: 11.6927 - val_acc: 0.5508
Epoch 37/100
236/236 [==============================] - 1s 3ms/sample - loss: 11.4793 - ac
c: 0.5381
1/1 [==============================] - 4s 4s/step - loss: 10.8319 - acc: 0.78
83 - val_loss: 11.4793 - val_acc: 0.5381
Epoch 38/100
236/236 [==============================] - 1s 3ms/sample - loss: 11.2510 - ac
c: 0.5805
1/1 [==============================] - 4s 4s/step - loss: 10.6308 - acc: 0.78
83 - val_loss: 11.2510 - val_acc: 0.5805
Epoch 39/100
236/236 [==============================] - 1s 3ms/sample - loss: 11.1499 - ac
c: 0.5551
1/1 [==============================] - 4s 4s/step - loss: 10.4002 - acc: 0.80
29 - val_loss: 11.1499 - val_acc: 0.5551
Epoch 40/100
236/236 [==============================] - 1s 3ms/sample - loss: 10.9399 - ac
c: 0.5466
1/1 [==============================] - 4s 4s/step - loss: 10.2108 - acc: 0.82
18 - val_loss: 10.9399 - val_acc: 0.5466
Epoch 41/100
236/236 [==============================] - 0s 2ms/sample - loss: 10.7843 - ac
c: 0.5551
1/1 [==============================] - 4s 4s/step - loss: 9.9921 - acc: 0.826
0 - val_loss: 10.7843 - val_acc: 0.5551
Epoch 42/100
236/236 [==============================] - 1s 3ms/sample - loss: 10.5703 - ac
c: 0.5636
1/1 [==============================] - 5s 5s/step - loss: 9.9108 - acc: 0.784
1 - val_loss: 10.5703 - val_acc: 0.5636
Epoch 43/100
236/236 [==============================] - 1s 3ms/sample - loss: 10.4225 - ac
c: 0.5678
1/1 [==============================] - 4s 4s/step - loss: 9.7093 - acc: 0.815
5 - val_loss: 10.4225 - val_acc: 0.5678
Epoch 44/100
236/236 [==============================] - 1s 3ms/sample - loss: 10.2858 - ac
c: 0.5551
1/1 [==============================] - 4s 4s/step - loss: 9.5221 - acc: 0.819
7 - val_loss: 10.2858 - val_acc: 0.5551
Epoch 45/100
236/236 [==============================] - 1s 3ms/sample - loss: 10.1342 - ac
c: 0.5466
1/1 [==============================] - 4s 4s/step - loss: 9.3630 - acc: 0.844
9 - val_loss: 10.1342 - val_acc: 0.5466
Epoch 46/100
236/236 [==============================] - 1s 3ms/sample - loss: 9.9628 - ac
c: 0.5381
```

```
1/1 [==============================] - 4s 4s/step - loss: 9.2264 - acc: 0.819
7 - val_loss: 9.9628 - val_acc: 0.5381
Epoch 47/100
236/236 [==============================] - 1s 3ms/sample - loss: 9.7784 - ac
c: 0.5593
1/1 [==============================] - 4s 4s/step - loss: 9.0681 - acc: 0.828
1 - val_loss: 9.7784 - val_acc: 0.5593
Epoch 48/100
236/236 [==============================] - 1s 3ms/sample - loss: 9.6620 - ac
c: 0.5847
1/1 [==============================] - 5s 5s/step - loss: 8.9047 - acc: 0.826
0 - val_loss: 9.6620 - val_acc: 0.5847
Epoch 49/100
236/236 [==============================] - 1s 3ms/sample - loss: 9.5614 - ac
c: 0.5593
1/1 [==============================] - 4s 4s/step - loss: 8.7983 - acc: 0.830
2 - val_loss: 9.5614 - val_acc: 0.5593
Epoch 50/100
236/236 [==============================] - 1s 3ms/sample - loss: 9.4227 - ac
c: 0.5847
1/1 [==============================] - 4s 4s/step - loss: 8.6614 - acc: 0.821
8 - val_loss: 9.4227 - val_acc: 0.5847
Epoch 51/100
236/236 [==============================] - 1s 3ms/sample - loss: 9.2827 - ac
c: 0.5636
1/1 [==============================] - 5s 5s/step - loss: 8.5511 - acc: 0.802
9 - val_loss: 9.2827 - val_acc: 0.5636
Epoch 52/100
236/236 [==============================] - 1s 3ms/sample - loss: 9.1809 - ac
c: 0.5678
1/1 [==============================] - 4s 4s/step - loss: 8.3947 - acc: 0.849
1 - val_loss: 9.1809 - val_acc: 0.5678
Epoch 53/100
236/236 [==============================] - 1s 3ms/sample - loss: 9.1138 - ac
c: 0.5466
1/1 [==============================] - 4s 4s/step - loss: 8.2952 - acc: 0.849
1 - val_loss: 9.1138 - val_acc: 0.5466
Epoch 54/100
236/236 [==============================] - 1s 3ms/sample - loss: 9.0582 - ac
c: 0.5212
1/1 [==============================] - 4s 4s/step - loss: 8.1852 - acc: 0.840
7 - val_loss: 9.0582 - val_acc: 0.5212
Epoch 55/100
236/236 [==============================] - 1s 3ms/sample - loss: 8.9583 - ac
c: 0.5636
1/1 [==============================] - 4s 4s/step - loss: 8.1143 - acc: 0.821
8 - val_loss: 8.9583 - val_acc: 0.5636
Epoch 56/100
236/236 [==============================] - 1s 3ms/sample - loss: 8.8797 - ac
c: 0.5508
1/1 [==============================] - 4s 4s/step - loss: 8.0041 - acc: 0.832
3 - val_loss: 8.8797 - val_acc: 0.5508
Epoch 57/100
236/236 [==============================] - 1s 2ms/sample - loss: 8.7540 - ac
c: 0.5636
1/1 [==============================] - 4s 4s/step - loss: 7.9364 - acc: 0.800
8 - val_loss: 8.7540 - val_acc: 0.5636
```

```
Epoch 58/100
236/236 [==============================] - 1s 2ms/sample - loss: 8.6521 - ac
c: 0.5593
1/1 [==============================] - 4s 4s/step - loss: 7.8355 - acc: 0.815
5 - val_loss: 8.6521 - val_acc: 0.5593
Epoch 59/100
236/236 [==============================] - 1s 3ms/sample - loss: 8.5620 - ac
c: 0.5593
1/1 [==============================] - 4s 4s/step - loss: 7.7053 - acc: 0.830
2 - val_loss: 8.5620 - val_acc: 0.5593
Epoch 60/100
236/236 [==============================] - 1s 3ms/sample - loss: 8.5303 - ac
c: 0.5381
1/1 [==============================] - 4s 4s/step - loss: 7.6147 - acc: 0.844
9 - val_loss: 8.5303 - val_acc: 0.5381
Epoch 61/100
236/236 [==============================] - 1s 3ms/sample - loss: 8.5018 - ac
c: 0.5339
1/1 [==============================] - 4s 4s/step - loss: 7.5783 - acc: 0.807
1 - val_loss: 8.5018 - val_acc: 0.5339
Epoch 62/100
236/236 [==============================] - 1s 3ms/sample - loss: 8.3359 - ac
c: 0.5593
1/1 [==============================] - 4s 4s/step - loss: 7.5463 - acc: 0.798
7 - val_loss: 8.3359 - val_acc: 0.5593
Epoch 63/100
236/236 [==============================] - 1s 3ms/sample - loss: 8.2272 - ac
c: 0.5381
1/1 [==============================] - 4s 4s/step - loss: 7.4203 - acc: 0.840
7 - val_loss: 8.2272 - val_acc: 0.5381
Epoch 64/100
236/236 [==============================] - 1s 2ms/sample - loss: 8.2067 - ac
c: 0.5593
1/1 [==============================] - 4s 4s/step - loss: 7.4000 - acc: 0.800
8 - val_loss: 8.2067 - val_acc: 0.5593
Epoch 65/100
236/236 [==============================] - 1s 3ms/sample - loss: 8.2436 - ac
c: 0.5381
1/1 [==============================] - 4s 4s/step - loss: 7.2934 - acc: 0.798
7 - val_loss: 8.2436 - val_acc: 0.5381
Epoch 66/100
236/236 [==============================] - 1s 3ms/sample - loss: 8.0825 - ac
c: 0.5508
1/1 [==============================] - 5s 5s/step - loss: 7.2862 - acc: 0.775
7 - val_loss: 8.0825 - val_acc: 0.5508
Epoch 67/100
236/236 [==============================] - 1s 3ms/sample - loss: 8.0425 - ac
c: 0.5551
1/1 [==============================] - 5s 5s/step - loss: 7.1247 - acc: 0.821
8 - val_loss: 8.0425 - val_acc: 0.5551
Epoch 68/100
236/236 [==============================] - 1s 2ms/sample - loss: 8.0089 - ac
c: 0.5339
1/1 [==============================] - 4s 4s/step - loss: 7.1261 - acc: 0.800
8 - val_loss: 8.0089 - val_acc: 0.5339
Epoch 69/100
236/236 [==============================] - 1s 2ms/sample - loss: 8.0046 - ac
```

```
c: 0.5551
1/1 [==============================] - 4s 4s/step - loss: 7.0292 - acc: 0.830
2 - val_loss: 8.0046 - val_acc: 0.5551
Epoch 70/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.9557 - ac
c: 0.5720
1/1 [==============================] - 4s 4s/step - loss: 6.9305 - acc: 0.849
1 - val_loss: 7.9557 - val_acc: 0.5720
Epoch 71/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.9192 - ac
c: 0.5890
1/1 [==============================] - 4s 4s/step - loss: 6.9119 - acc: 0.817
6 - val_loss: 7.9192 - val_acc: 0.5890
Epoch 72/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.8718 - ac
c: 0.5763
1/1 [==============================] - 4s 4s/step - loss: 6.9137 - acc: 0.811
3 - val_loss: 7.8718 - val_acc: 0.5763
Epoch 73/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.8514 - ac
c: 0.5424
1/1 [==============================] - 4s 4s/step - loss: 6.8453 - acc: 0.813
4 - val_loss: 7.8514 - val_acc: 0.5424
Epoch 74/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.7285 - ac
c: 0.5424
1/1 [==============================] - 4s 4s/step - loss: 6.8059 - acc: 0.788
3 - val_loss: 7.7285 - val_acc: 0.5424
Epoch 75/100
236/236 [==============================] - 1s 4ms/sample - loss: 7.7076 - ac
c: 0.5424
1/1 [==============================] - 4s 4s/step - loss: 6.8157 - acc: 0.788
3 - val_loss: 7.7076 - val_acc: 0.5424
Epoch 76/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.6501 - ac
c: 0.5127
1/1 [==============================] - 4s 4s/step - loss: 6.7140 - acc: 0.811
3 - val_loss: 7.6501 - val_acc: 0.5127
Epoch 77/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.7290 - ac
c: 0.5424
1/1 [==============================] - 4s 4s/step - loss: 6.5929 - acc: 0.830
2 - val_loss: 7.7290 - val_acc: 0.5424
Epoch 78/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.5062 - ac
c: 0.5466
1/1 [==============================] - 5s 5s/step - loss: 6.6909 - acc: 0.752
6 - val_loss: 7.5062 - val_acc: 0.5466
Epoch 79/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.4775 - ac
c: 0.5805
1/1 [==============================] - 4s 4s/step - loss: 6.5402 - acc: 0.849
1 - val_loss: 7.4775 - val_acc: 0.5805
Epoch 80/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.3918 - ac
c: 0.5847
1/1 [==============================] - 4s 4s/step - loss: 6.5769 - acc: 0.792
```

```
5 - val_loss: 7.3918 - val_acc: 0.5847
Epoch 81/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.4829 - ac
c: 0.5551
1/1 [==============================] - 4s 4s/step - loss: 6.4756 - acc: 0.817
6 - val_loss: 7.4829 - val_acc: 0.5551
Epoch 82/100
236/236 [==============================] - 1s 2ms/sample - loss: 7.4484 - ac
c: 0.5381
1/1 [==============================] - 4s 4s/step - loss: 6.4918 - acc: 0.817
6 - val_loss: 7.4484 - val_acc: 0.5381
Epoch 83/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.3770 - ac
c: 0.5381
1/1 [==============================] - 4s 4s/step - loss: 6.4477 - acc: 0.805
0 - val_loss: 7.3770 - val_acc: 0.5381
Epoch 84/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.3421 - ac
c: 0.5381
1/1 [==============================] - 4s 4s/step - loss: 6.4254 - acc: 0.809
2 - val_loss: 7.3421 - val_acc: 0.5381
Epoch 85/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.3877 - ac
c: 0.5381
1/1 [==============================] - 4s 4s/step - loss: 6.3241 - acc: 0.838
6 - val_loss: 7.3877 - val_acc: 0.5381
Epoch 86/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.4475 - ac
c: 0.5127
1/1 [==============================] - 4s 4s/step - loss: 6.3387 - acc: 0.832
3 - val_loss: 7.4475 - val_acc: 0.5127
Epoch 87/100
236/236 [==============================] - 1s 2ms/sample - loss: 7.4534 - ac
c: 0.5339
1/1 [==============================] - 4s 4s/step - loss: 6.3400 - acc: 0.807
1 - val_loss: 7.4534 - val_acc: 0.5339
Epoch 88/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.3702 - ac
c: 0.5466
1/1 [==============================] - 4s 4s/step - loss: 6.2741 - acc: 0.811
3 - val_loss: 7.3702 - val_acc: 0.5466
Epoch 89/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.3258 - ac
c: 0.5508
1/1 [==============================] - 4s 4s/step - loss: 6.1855 - acc: 0.840
7 - val_loss: 7.3258 - val_acc: 0.5508
Epoch 90/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.1655 - ac
c: 0.5466
1/1 [==============================] - 4s 4s/step - loss: 6.1941 - acc: 0.838
6 - val_loss: 7.1655 - val_acc: 0.5466
Epoch 91/100
236/236 [==============================] - 1s 2ms/sample - loss: 7.2315 - ac
c: 0.5212
1/1 [==============================] - 4s 4s/step - loss: 6.1511 - acc: 0.832
3 - val_loss: 7.2315 - val_acc: 0.5212
Epoch 92/100
```

```
236/236 [==============================] - 0s 2ms/sample - loss: 7.2142 - ac
c: 0.5085
1/1 [==============================] - 4s 4s/step - loss: 6.1725 - acc: 0.794
5 - val_loss: 7.2142 - val_acc: 0.5085
Epoch 93/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.1464 - ac
c: 0.5085
1/1 [==============================] - 4s 4s/step - loss: 6.1161 - acc: 0.836
5 - val_loss: 7.1464 - val_acc: 0.5085
Epoch 94/100
236/236 [==============================] - 0s 2ms/sample - loss: 7.1302 - ac
c: 0.5424
1/1 [==============================] - 4s 4s/step - loss: 6.0432 - acc: 0.826
0 - val_loss: 7.1302 - val_acc: 0.5424
Epoch 95/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.0941 - ac
c: 0.5593
1/1 [==============================] - 4s 4s/step - loss: 6.0365 - acc: 0.840
7 - val_loss: 7.0941 - val_acc: 0.5593
Epoch 96/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.1593 - ac
c: 0.5339
1/1 [==============================] - 4s 4s/step - loss: 5.9892 - acc: 0.851
2 - val_loss: 7.1593 - val_acc: 0.5339
Epoch 97/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.0502 - ac
c: 0.5551
1/1 [==============================] - 4s 4s/step - loss: 5.9517 - acc: 0.823
9 - val_loss: 7.0502 - val_acc: 0.5551
Epoch 98/100
236/236 [==============================] - 1s 3ms/sample - loss: 6.9429 - ac
c: 0.5466
1/1 [==============================] - 4s 4s/step - loss: 5.9446 - acc: 0.849
1 - val_loss: 6.9429 - val_acc: 0.5466
Epoch 99/100
236/236 [==============================] - 1s 2ms/sample - loss: 6.9183 - ac
c: 0.5466
1/1 [==============================] - 4s 4s/step - loss: 5.9049 - acc: 0.840
7 - val_loss: 6.9183 - val_acc: 0.5466
Epoch 100/100
236/236 [==============================] - 1s 3ms/sample - loss: 7.0233 - ac
c: 0.5508
1/1 [==============================] - 4s 4s/step - loss: 5.8821 - acc: 0.828
1 - val_loss: 7.0233 - val_acc: 0.5508
```

Out[76]: <tensorflow.python.keras.callbacks.History at 0x7fd869164748>

In [33]:
```
score = model.evaluate(xtest, ytest_chan, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
pred = model.predict(xtest)
```

```
Test loss: 0.26396609564958995
Test accuracy: 0.970339
```

In [166]:
```
ytest_chan.T.shape
```

Out[166]:
```
(10, 236)
```

In [167]:
```
pred.T.shape
```

Out[167]:
```
(10, 236)
```

In [168]:
```
print("-Confusion matrix")
confuse = tf.math.confusion_matrix(np.argmax(ytest_chan,axis = 1), np.argmax(p
red, axis = 1), num_classes = 10)
print(confuse)
```

```
tf.Tensor(
[[10  1  3  3  0  2  4  1  2  0]
 [ 0 18  1  1  1  2  0  0  1  1]
 [ 1  0 15  1  1  4  1  0  1  0]
 [ 0  0  1 14  1  6  0  1  0  0]
 [ 0  1  3  0 13  2  0  4  1  1]
 [ 0  0  3  2  0 12  4  0  1  0]
 [ 0  0  0  0  1  5 15  0  3  0]
 [ 0  2  0  3  1  3  3 15  0  1]
 [ 0  0  2  1  0  3  1  0 13  0]
 [ 0  3  0  0  2  1  0  3  1  9]], shape=(10, 10), dtype=int32)
```

In [169]:
```
np.argmax(pred,axis = 1)
```

Out[169]:
```
array([9, 5, 7, 2, 6, 2, 5, 1, 4, 5, 2, 4, 2, 5, 0, 6, 4, 5, 1, 6, 5, 3,
       5, 9, 5, 8, 4, 7, 5, 2, 6, 5, 6, 7, 7, 8, 0, 5, 7, 5, 7, 5, 5, 5,
       6, 3, 9, 9, 6, 1, 1, 8, 8, 4, 2, 3, 8, 5, 1, 5, 1, 2, 7, 5, 1, 3,
       6, 4, 2, 8, 0, 0, 9, 7, 7, 9, 5, 8, 6, 3, 3, 7, 6, 8, 0, 2, 8, 7,
       6, 7, 2, 1, 1, 6, 9, 2, 2, 7, 5, 4, 8, 3, 6, 2, 0, 5, 0, 2, 7, 4,
       8, 6, 2, 4, 2, 2, 3, 6, 8, 1, 3, 1, 8, 4, 2, 5, 7, 7, 3, 8, 3, 1,
       5, 1, 3, 6, 1, 3, 9, 7, 6, 0, 5, 3, 8, 6, 5, 9, 8, 7, 0, 1, 6, 2,
       3, 5, 9, 3, 6, 1, 7, 1, 5, 5, 3, 3, 6, 5, 5, 5, 5, 1, 1, 3, 5, 4,
       2, 8, 4, 6, 2, 4, 4, 0, 8, 8, 2, 6, 6, 2, 5, 7, 1, 4, 7, 4, 9, 1,
       2, 1, 1, 4, 6, 5, 3, 9, 5, 3, 5, 0, 3, 1, 4, 5, 6, 2, 6, 8, 8, 4,
       7, 8, 7, 2, 2, 2, 7, 3, 1, 4, 3, 6, 5, 3, 8, 5])
```

In [170]:
```
np.argmax(ytest_chan, axis = 1)
```

Out[170]:
```
array([9, 5, 4, 2, 6, 2, 8, 9, 2, 5, 2, 4, 5, 9, 0, 6, 9, 0, 1, 6, 8, 3,
       6, 7, 1, 8, 4, 7, 3, 2, 0, 5, 6, 3, 7, 2, 0, 3, 7, 5, 7, 5, 7, 5,
       8, 0, 1, 9, 6, 0, 9, 8, 8, 4, 2, 3, 8, 5, 1, 6, 1, 3, 7, 3, 1, 3,
       7, 4, 8, 1, 0, 0, 4, 4, 0, 9, 5, 8, 6, 3, 8, 7, 6, 0, 0, 8, 8, 7,
       6, 7, 4, 9, 1, 0, 9, 2, 2, 7, 3, 4, 9, 3, 6, 0, 0, 8, 2, 2, 9, 4,
       8, 7, 5, 4, 2, 1, 7, 6, 8, 1, 3, 1, 0, 6, 5, 2, 9, 7, 3, 8, 3, 1,
       6, 7, 3, 7, 1, 5, 9, 7, 5, 0, 0, 1, 8, 0, 2, 9, 4, 7, 0, 1, 0, 4,
       7, 4, 9, 3, 6, 1, 4, 1, 6, 5, 3, 0, 6, 3, 7, 6, 5, 1, 1, 3, 1, 7,
       4, 8, 4, 6, 2, 4, 9, 0, 8, 6, 0, 6, 5, 2, 2, 9, 7, 4, 7, 4, 9, 1,
       0, 1, 1, 3, 5, 3, 3, 9, 5, 7, 4, 0, 5, 1, 1, 5, 2, 2, 6, 5, 8, 4,
       4, 6, 7, 2, 2, 2, 7, 2, 4, 4, 0, 5, 7, 3, 6, 2])
```

Interesting the model is seeing everything in the validation set as one particular class overfiting----- high variance options

1 more data ? 2 regularization 3 more appropriate architecture

In [36]:
```
### Saving model
model_json = model.to_json()
with open("Models/model{}_json.json".format(idc), "w") as j_file:
    j_file.write(model_json)
```

In [37]:
```
### Saving model's weights
model.save_weights("Models/model{}.h5".format(idc))
```