Gesture Recognition Experiment Log

**Fixed for All Experiments**

num\_train\_sequences = 663

num\_val\_sequences   = 100

num\_classes=5

loss='categorical\_crossentropy',

metrics=['categorical\_accuracy']

**CNN Experiments**

Experiments 1

**Jupyter\_file name:** Neural\_Nets\_Project\_Starter\_Code1.1

Decision/ Explanation:

Image Size = 100x100

Batch Size = 5

img\_idx = random 10 images from each sequence

num\_epochs = 5

Take only 15 videos for training

**Architecture**

model.add(Conv3D(32, (2, 3, 3), padding='same', input\_shape=(imgs\_from\_1video, imsize[0],imsize[1],3)))

model.add(Activation('relu'))

model.add(Conv3D(32, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.25))

model.add(Conv3D(64, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(Conv3D(64, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(512))

model.add(Activation('relu'))

model.add(Dropout(0.5))

model.add(Dense(num\_classes))

model.add(Activation('softmax'))

Resutls

Training Accuracy: 20%

Experiments 2

**Jupyter\_file name:** Neural\_Nets\_Project\_Starter\_Code1.2

**Decision/ Explanation:**

Image Size = 80x80

Batch Size = 10

img\_idx = random 15 images from each sequence

num\_epochs = 10

Take only 15 videos for training

Results

Training Accuracy: 20%

Experiments 3

**Jupyter\_file name:** Neural\_Nets\_Project\_Starter\_Code1.3

**Decision/ Explanation:**

Image Size = 80x80

Batch Size = 10

img\_idx = random 15 images from each sequence

num\_epochs = 10

Take only 15 videos for training

Results

Training Accuracy: 20%

Experiments 4

**Jupyter\_file name:** Neural\_Nets\_Project\_Starter\_Code1.3

**Decision/ Explanation:**

Add CONV3D 3rd Layer of 128 neurons

**Architecture**

model.add(Conv3D(32, (2, 3, 3), padding='same', input\_shape=(imgs\_from\_1video, imsize[0],imsize[1],3)))

model.add(Activation('relu'))

model.add(Conv3D(32, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.25))

model.add(Conv3D(64, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(Conv3D(64, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.25))

model.add(Conv3D(128, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(Conv3D(128, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(512))

model.add(Activation('relu'))

model.add(Dropout(0.5))

model.add(Dense(num\_classes))

model.add(Activation('softmax'))

Results

Training Accuracy: 21%

Experiments 5

**Jupyter\_file name:** Neural\_Nets\_Project\_Starter\_Code1

**Decision/ Explanation:**

num\_epochs = 50

Other Parameters

Image Size = 128x128

Batch Size = 60

img\_idx = np.array( range(0,30,3))

num\_epochs = 50

optimiser = "adam"

Normalization

steps\_per\_epoch = int(num\_train\_sequences/batch\_size) => 12

validation\_steps = int(num\_val\_sequences/batch\_size)  => 2

**Results**

Experiments - 6

Jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code2

**Decision/ Explanation:**

Batch Size = 40

num\_epochs = 100

**Other Parameters**

Image Size = 128x128

Batch Size = 40

img\_idx = np.array( range(0,30,3)) #10 images

num\_epochs = 100

optimiser = "adam"

steps\_per\_epoch = int(num\_train\_sequences/batch\_size) => 17

validation\_steps = int(num\_val\_sequences/batch\_size)  => 2

**Architecture**

model.add(Conv3D(32, (2, 3, 3), padding='same', input\_shape=(imgs\_from\_1video, imsize[0],imsize[1],3)))

model.add(Activation('relu'))

model.add(Conv3D(32, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.25))

model.add(Conv3D(64, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(Conv3D(64, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.25))

model.add(Conv3D(128, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(Conv3D(128, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(512))

model.add(Activation('relu'))

model.add(Dropout(0.5))

model.add(Dense(num\_classes))

model.add(Activation('softmax'))

Architecture

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv3d\_1 (Conv3D) (None, 10, 128, 128, 32) 1760

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_1 (Activation) (None, 10, 128, 128, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_2 (Conv3D) (None, 10, 128, 128, 32) 18464

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_2 (Activation) (None, 10, 128, 128, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling3d\_1 (MaxPooling3 (None, 5, 64, 64, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_1 (Dropout) (None, 5, 64, 64, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_3 (Conv3D) (None, 5, 64, 64, 64) 36928

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_3 (Activation) (None, 5, 64, 64, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_4 (Conv3D) (None, 5, 64, 64, 64) 73792

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_4 (Activation) (None, 5, 64, 64, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling3d\_2 (MaxPooling3 (None, 2, 32, 32, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_2 (Dropout) (None, 2, 32, 32, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_5 (Conv3D) (None, 2, 32, 32, 128) 147584

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_5 (Activation) (None, 2, 32, 32, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_6 (Conv3D) (None, 2, 32, 32, 128) 295040

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_6 (Activation) (None, 2, 32, 32, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling3d\_3 (MaxPooling3 (None, 1, 16, 16, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_3 (Dropout) (None, 1, 16, 16, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_1 (Flatten) (None, 32768) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 512) 16777728

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_7 (Activation) (None, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_4 (Dropout) (None, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_2 (Dense) (None, 5) 2565

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_8 (Activation) (None, 5) 0

=================================================================

Total params: 17,353,861

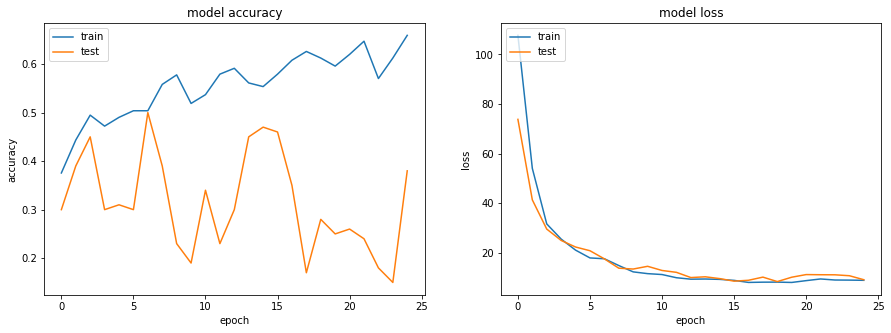
Trainable params: 17,353,861

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

None

**Results**



Experiments - 7

**Jupyter\_file name:** Neural\_Nets\_Project\_Starter\_Code3

**Decision/ Explanation:**

num\_epochs = 50 (reduced)

Added BatchNormalization in Architecture

**Other Parameters**

Image Size = 128x128

Batch Size = 40

img\_idx = np.array( range(0,30,3)) #10 images

num\_epochs = 50 (reduced)

optimiser = "adam"

steps\_per\_epoch = int(num\_train\_sequences/batch\_size) => 17

validation\_steps = int(num\_val\_sequences/batch\_size)  => 2

**Architecture**

model = Sequential()

model.add(Conv3D(32, (2, 3, 3), padding='same', input\_shape=(imgs\_from\_1video, imsize[0],imsize[1],3)))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(Conv3D(32, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.25))

model.add(Conv3D(64, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(Conv3D(64, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.25))

model.add(Conv3D(128, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(Conv3D(128, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(512))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(Dropout(0.5))

model.add(Dense(num\_classes))

model.add(Activation('softmax'))

#write your model here

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv3d\_1 (Conv3D) (None, 10, 128, 128, 32) 1760

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_1 (Activation) (None, 10, 128, 128, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_1 (Batch (None, 10, 128, 128, 32) 128

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_2 (Conv3D) (None, 10, 128, 128, 32) 18464

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_2 (Activation) (None, 10, 128, 128, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_2 (Batch (None, 10, 128, 128, 32) 128

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling3d\_1 (MaxPooling3 (None, 5, 64, 64, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_1 (Dropout) (None, 5, 64, 64, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_3 (Conv3D) (None, 5, 64, 64, 64) 36928

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_3 (Activation) (None, 5, 64, 64, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_3 (Batch (None, 5, 64, 64, 64) 256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_4 (Conv3D) (None, 5, 64, 64, 64) 73792

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_4 (Activation) (None, 5, 64, 64, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_4 (Batch (None, 5, 64, 64, 64) 256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling3d\_2 (MaxPooling3 (None, 2, 32, 32, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_2 (Dropout) (None, 2, 32, 32, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_5 (Conv3D) (None, 2, 32, 32, 128) 147584

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_5 (Activation) (None, 2, 32, 32, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_5 (Batch (None, 2, 32, 32, 128) 512

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_6 (Conv3D) (None, 2, 32, 32, 128) 295040

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_6 (Activation) (None, 2, 32, 32, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_6 (Batch (None, 2, 32, 32, 128) 512

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling3d\_3 (MaxPooling3 (None, 1, 16, 16, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_3 (Dropout) (None, 1, 16, 16, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_1 (Flatten) (None, 32768) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 512) 16777728

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_7 (Activation) (None, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_7 (Batch (None, 512) 2048

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_4 (Dropout) (None, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_2 (Dense) (None, 5) 2565

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_8 (Activation) (None, 5) 0

=================================================================

Total params: 17,357,701

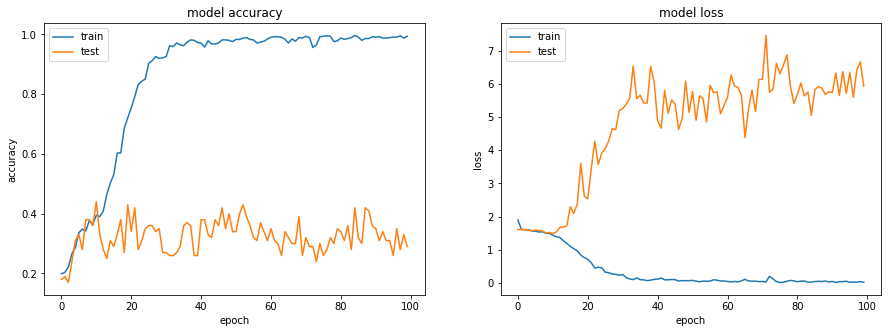
Trainable params: 17,355,781

Non-trainable params: 1,920

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

None

**Results**



0\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*0

Experiments - 8

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code4

**Decision/ Explanation:**

num\_epochs = 25 (reduced)

Introduced Kernal Regularizer. kernel\_regularizer=l2(0.01)

model.add(Dense(512, kernel\_regularizer=l2(0.001)))

**Other Parameters**

Image Size = 128x128

Batch Size = 40

img\_idx = np.array( range(0,30,3)) #10 images

num\_epochs = 25 (reduced)

optimiser = "adam"

steps\_per\_epoch = int(num\_train\_sequences/batch\_size) => 17

validation\_steps = int(num\_val\_sequences/batch\_size)  => 2

**Architecture**

model.add(Conv3D(32, (2, 3, 3), kernel\_regularizer=l2(0.01), bias\_regularizer=l2(0.01), padding='same',  input\_shape=(imgs\_from\_1video, imsize[0],imsize[1],3)))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(Conv3D(32, (2, 3, 3), kernel\_regularizer=l2(0.01), bias\_regularizer=l2(0.01), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.25))

model.add(Conv3D(64, (2, 3, 3), kernel\_regularizer=l2(0.01), bias\_regularizer=l2(0.01), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(Conv3D(64, (2, 3, 3), kernel\_regularizer=l2(0.01), bias\_regularizer=l2(0.01), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.25))

model.add(Conv3D(128, (2, 3, 3), kernel\_regularizer=l2(0.01), bias\_regularizer=l2(0.01), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(Conv3D(128, (2, 3, 3), kernel\_regularizer=l2(0.01), bias\_regularizer=l2(0.01), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(512, kernel\_regularizer=l2(0.001)))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(Dropout(0.5))

model.add(Dense(num\_classes))

model.add(Activation('softmax'))

#write your model here

**Model Summary**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv3d\_1 (Conv3D) (None, 10, 128, 128, 32) 1760

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_1 (Activation) (None, 10, 128, 128, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_1 (Batch (None, 10, 128, 128, 32) 128

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_2 (Conv3D) (None, 10, 128, 128, 32) 18464

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_2 (Activation) (None, 10, 128, 128, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_2 (Batch (None, 10, 128, 128, 32) 128

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling3d\_1 (MaxPooling3 (None, 5, 64, 64, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_1 (Dropout) (None, 5, 64, 64, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_3 (Conv3D) (None, 5, 64, 64, 64) 36928

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_3 (Activation) (None, 5, 64, 64, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_3 (Batch (None, 5, 64, 64, 64) 256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_4 (Conv3D) (None, 5, 64, 64, 64) 73792

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_4 (Activation) (None, 5, 64, 64, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_4 (Batch (None, 5, 64, 64, 64) 256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling3d\_2 (MaxPooling3 (None, 2, 32, 32, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_2 (Dropout) (None, 2, 32, 32, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_5 (Conv3D) (None, 2, 32, 32, 128) 147584

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_5 (Activation) (None, 2, 32, 32, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_5 (Batch (None, 2, 32, 32, 128) 512

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_6 (Conv3D) (None, 2, 32, 32, 128) 295040

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_6 (Activation) (None, 2, 32, 32, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_6 (Batch (None, 2, 32, 32, 128) 512

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling3d\_3 (MaxPooling3 (None, 1, 16, 16, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_3 (Dropout) (None, 1, 16, 16, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_1 (Flatten) (None, 32768) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 512) 16777728

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_7 (Activation) (None, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_7 (Batch (None, 512) 2048

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_4 (Dropout) (None, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_2 (Dense) (None, 5) 2565

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_8 (Activation) (None, 5) 0

=================================================================

Total params: 17,357,701

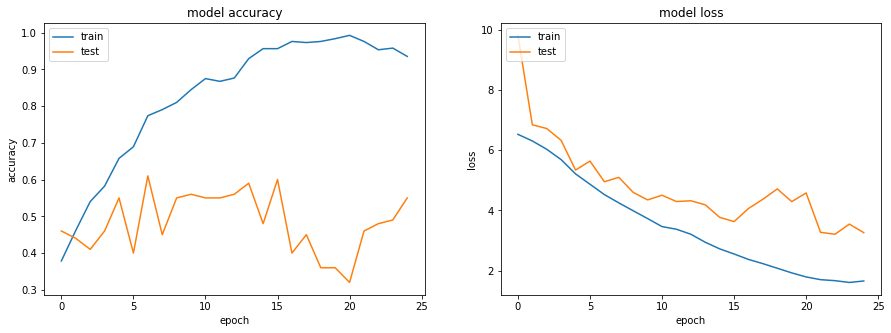
Trainable params: 17,355,781

Non-trainable params: 1,920

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

None

**Result**



0\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*0

Experiments - 9

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code5

**Decision/ Explanation:**

change kernel size. 3x3 => 4x4

increaes drop out. 25% => 30%

kernel\_regularizer=l2(0.1), bias\_regularizer=l2(0.1)

model.add(Dense(512, kernel\_regularizer=l2(0.1)))

**Other Parameters**

Image Size = 128x128

Batch Size = 40

img\_idx = np.array( range(0,30,3)) #10 images

num\_epochs = 25 (reduced batch size)

optimiser = "adam"

steps\_per\_epoch = int(num\_train\_sequences/batch\_size) => 17

validation\_steps = int(num\_val\_sequences/batch\_size)  => 2

Architecture

model.add(Conv3D(32, (2, 4, 4), padding='same', input\_shape=(imgs\_from\_1video, imsize[0],imsize[1],3)))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(Conv3D(32, (2, 4, 4), kernel\_regularizer=l2(0.1), bias\_regularizer=l2(0.1), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.30))

model.add(Conv3D(64, (2, 4, 4), kernel\_regularizer=l2(0.1), bias\_regularizer=l2(0.1), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(Conv3D(64, (2, 4, 4), kernel\_regularizer=l2(0.1), bias\_regularizer=l2(0.1), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.30))

model.add(Conv3D(128, (2, 4, 4), kernel\_regularizer=l2(0.1), bias\_regularizer=l2(0.1), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(Conv3D(128, (2, 4, 4), kernel\_regularizer=l2(0.1), bias\_regularizer=l2(0.1), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.30))

model.add(Flatten())

model.add(Dense(512, kernel\_regularizer=l2(0.1)))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(Dropout(0.6))

model.add(Dense(num\_classes))

model.add(Activation('softmax'))

#write your model here

Model Summary

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv3d\_1 (Conv3D) (None, 10, 128, 128, 32) 3104

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_1 (Activation) (None, 10, 128, 128, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_1 (Batch (None, 10, 128, 128, 32) 128

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_2 (Conv3D) (None, 10, 128, 128, 32) 32800

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_2 (Activation) (None, 10, 128, 128, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_2 (Batch (None, 10, 128, 128, 32) 128

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling3d\_1 (MaxPooling3 (None, 5, 64, 64, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_1 (Dropout) (None, 5, 64, 64, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_3 (Conv3D) (None, 5, 64, 64, 64) 65600

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_3 (Activation) (None, 5, 64, 64, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_3 (Batch (None, 5, 64, 64, 64) 256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_4 (Conv3D) (None, 5, 64, 64, 64) 131136

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_4 (Activation) (None, 5, 64, 64, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_4 (Batch (None, 5, 64, 64, 64) 256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling3d\_2 (MaxPooling3 (None, 2, 32, 32, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_2 (Dropout) (None, 2, 32, 32, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_5 (Conv3D) (None, 2, 32, 32, 128) 262272

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_5 (Activation) (None, 2, 32, 32, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_5 (Batch (None, 2, 32, 32, 128) 512

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_6 (Conv3D) (None, 2, 32, 32, 128) 524416

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_6 (Activation) (None, 2, 32, 32, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_6 (Batch (None, 2, 32, 32, 128) 512

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling3d\_3 (MaxPooling3 (None, 1, 16, 16, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_3 (Dropout) (None, 1, 16, 16, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_1 (Flatten) (None, 32768) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 512) 16777728

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_7 (Activation) (None, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_7 (Batch (None, 512) 2048

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_4 (Dropout) (None, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_2 (Dense) (None, 5) 2565

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_8 (Activation) (None, 5) 0

=================================================================

Total params: 17,803,461

Trainable params: 17,801,541

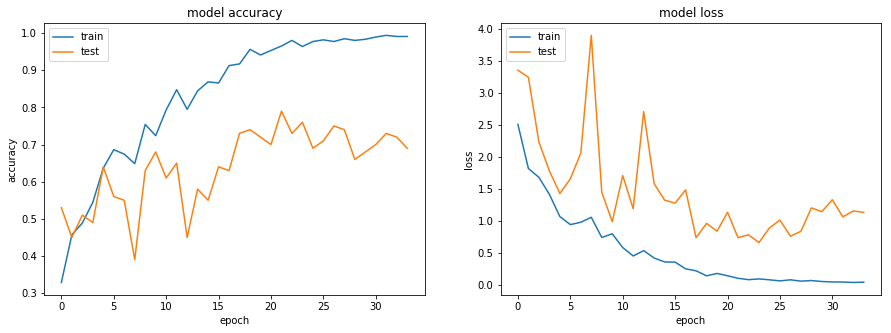
Non-trainable params: 1,920

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

None

**Results**

Training Accuracy dropped significantly and no improvement in val accuracy



0\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*0

Experiments - 9

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code6

**Decision/ Explanation:**

num\_epochs = 60

Add kernel to first conv3d layer. kernel\_regularizer=l2(0.1)

model.add(Dense(512, kernel\_regularizer=l2(0.001)))

**Other Parameters from Previous Experiment**

change kernel size. 3x3 => 4x4

kernel\_regularizer=l2(0.1), bias\_regularizer=l2(0.1)

Image Size = 128x128

Batch Size = 40

img\_idx = np.array( range(0,30,3)) #10 images

num\_epochs = 25 (reduced batch size)

optimiser = "adam"

steps\_per\_epoch = int(num\_train\_sequences/batch\_size) => 17

validation\_steps = int(num\_val\_sequences/batch\_size)  => 2

**Architecture**

model = Sequential()

model.add(Conv3D(32, (2, 4, 4), kernel\_regularizer=l2(0.1), padding='same',

                 input\_shape=(imgs\_from\_1video, imsize[0],imsize[1],3)))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(Conv3D(32, (2, 4, 4), kernel\_regularizer=l2(0.1), bias\_regularizer=l2(0.1), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.30))

model.add(Conv3D(64, (2, 4, 4), kernel\_regularizer=l2(0.1), bias\_regularizer=l2(0.1), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(Conv3D(64, (2, 4, 4), kernel\_regularizer=l2(0.1), bias\_regularizer=l2(0.1), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.30))

model.add(Conv3D(128, (2, 4, 4), kernel\_regularizer=l2(0.1), bias\_regularizer=l2(0.1), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(Conv3D(128, (2, 4, 4), kernel\_regularizer=l2(0.1), bias\_regularizer=l2(0.1), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.30))

model.add(Flatten())

model.add(Dense(512, kernel\_regularizer=l2(0.001)))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(Dropout(0.6))

model.add(Dense(num\_classes))

model.add(Activation('softmax'))

#write your model here

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv3d\_1 (Conv3D) (None, 10, 128, 128, 32) 3104

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_1 (Activation) (None, 10, 128, 128, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_1 (Batch (None, 10, 128, 128, 32) 128

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_2 (Conv3D) (None, 10, 128, 128, 32) 32800

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_2 (Activation) (None, 10, 128, 128, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_2 (Batch (None, 10, 128, 128, 32) 128

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling3d\_1 (MaxPooling3 (None, 5, 64, 64, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_1 (Dropout) (None, 5, 64, 64, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_3 (Conv3D) (None, 5, 64, 64, 64) 65600

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_3 (Activation) (None, 5, 64, 64, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_3 (Batch (None, 5, 64, 64, 64) 256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_4 (Conv3D) (None, 5, 64, 64, 64) 131136

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_4 (Activation) (None, 5, 64, 64, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_4 (Batch (None, 5, 64, 64, 64) 256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling3d\_2 (MaxPooling3 (None, 2, 32, 32, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_2 (Dropout) (None, 2, 32, 32, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_5 (Conv3D) (None, 2, 32, 32, 128) 262272

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_5 (Activation) (None, 2, 32, 32, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_5 (Batch (None, 2, 32, 32, 128) 512

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv3d\_6 (Conv3D) (None, 2, 32, 32, 128) 524416

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_6 (Activation) (None, 2, 32, 32, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_6 (Batch (None, 2, 32, 32, 128) 512

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling3d\_3 (MaxPooling3 (None, 1, 16, 16, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_3 (Dropout) (None, 1, 16, 16, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_1 (Flatten) (None, 32768) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 512) 16777728

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_7 (Activation) (None, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_7 (Batch (None, 512) 2048

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_4 (Dropout) (None, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_2 (Dense) (None, 5) 2565

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_8 (Activation) (None, 5) 0

=================================================================

Total params: 17,803,461

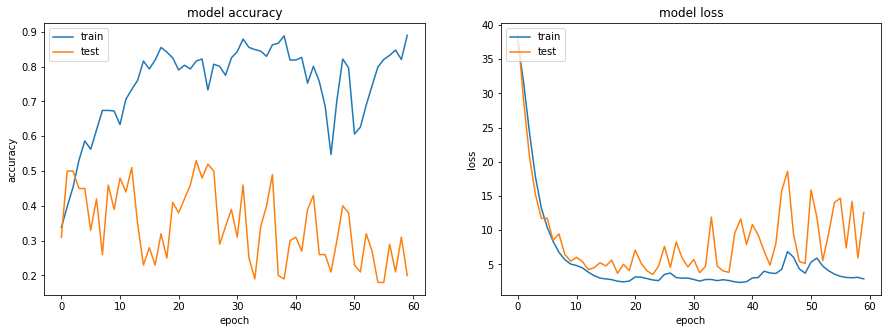
Trainable params: 17,801,541

Non-trainable params: 1,920

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

None

**Results**



0\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*0

Experiments - 10

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code7

**Decision/ Explanation:**

num\_epochs = 25

enable saving models

**Other Parameters from Previous Experiment**

kernel size. 3x3

Image Size = 128x128

Batch Size = 40

img\_idx = np.array( range(0,30,3)) #10 images

num\_epochs = 25 (reduced batch size)

optimiser = "adam"

steps\_per\_epoch = int(num\_train\_sequences/batch\_size) => 17

validation\_steps = int(num\_val\_sequences/batch\_size)  => 2

**Architecture**

model.add(Conv3D(32, (2, 3, 3), padding='same',

input\_shape=(imgs\_from\_1video, imsize[0],imsize[1],3)))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(Conv3D(32, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.25))

model.add(Conv3D(64, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(Conv3D(64, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.25))

model.add(Conv3D(128, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(Conv3D(128, (2, 3, 3), padding='same'))

model.add(Activation('relu'))

model.add(BatchNormalization())

model.add(MaxPooling3D(pool\_size=(2, 2, 2)))

model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(512))

model.add(Activation('relu'))

model.add(BatchNormalization())

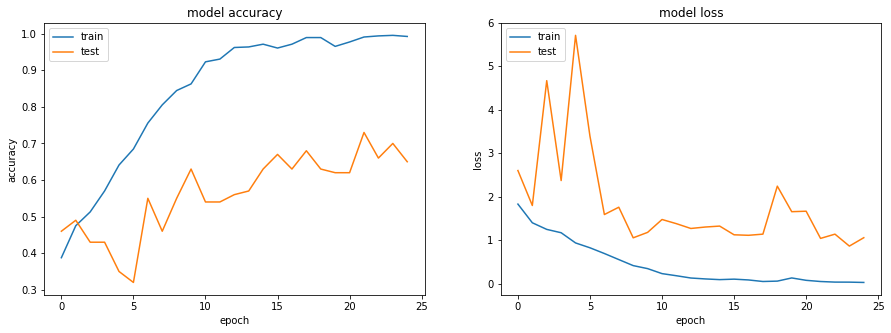
model.add(Dropout(0.5))

model.add(Dense(num\_classes))

model.add(Activation('softmax'))

#write your model here

**Results**



**CNN+RNN**

Experiments – 11 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code8

**Decision/ Explanation:**

num\_epochs = 25

disable saving models

**Other Parameters from Previous Experiment**

kernel size. 3x3

Image Size = 128x128

Batch Size = 40

img\_idx = np.array( range(0,30,3)) #10 images

num\_epochs = 25 (reduced batch size)

optimiser = "adam"

steps\_per\_epoch = int(num\_train\_sequences/batch\_size) => 17

validation\_steps = int(num\_val\_sequences/batch\_size)  => 2

**Architecture**

#### Set1

model.add(TimeDistributed(Conv2D(32, (5, 5), strides=(2, 2), activation='relu',

padding='same'), input\_shape=(imgs\_from\_1video, imsize[0],imsize[1], 3)))

model.add(TimeDistributed(Conv2D(32, (3,3), kernel\_initializer="he\_normal", activation='relu')))

#model.add(BatchNormalization())

model.add(TimeDistributed( MaxPooling2D((2, 2), strides=(2, 2))) )

model.add(Dropout(0.25))

#### Set2

model.add(TimeDistributed(Conv2D(64, (3,3), padding='same', activation='relu')))

model.add(TimeDistributed(Conv2D(64, (3,3), padding='same', activation='relu')))

#model.add(BatchNormalization())

model.add(TimeDistributed( MaxPooling2D(pool\_size=(2, 2))) )

model.add(Dropout(0.25))

#### Set3

model.add(TimeDistributed(Conv2D(124, (3,3), padding='same', activation='relu')))

model.add(TimeDistributed(Conv2D(124, (3,3), padding='same', activation='relu')))

#model.add(BatchNormalization())

model.add(TimeDistributed( MaxPooling2D(pool\_size=(2, 2))) )

model.add(Dropout(0.25))

#### Set4

model.add(TimeDistributed(Conv2D(256, (3,3), padding='same', activation='relu')))

model.add(TimeDistributed(Conv2D(256, (3,3), padding='same', activation='relu')))

#model.add(BatchNormalization())

model.add(TimeDistributed( MaxPooling2D(pool\_size=(2, 2))) )

model.add(Dropout(0.25))

#### Set3

model.add(TimeDistributed(Conv2D(512, (3,3), padding='same', activation='relu')))

model.add(TimeDistributed(Conv2D(512, (3,3), padding='same', activation='relu')))

#model.add(BatchNormalization())

model.add(TimeDistributed( MaxPooling2D(pool\_size=(2, 2))))

model.add(Dropout(0.25))

model.add(TimeDistributed(Flatten()))

model.add(Dropout(0.5))

model.add(GRU(256, return\_sequences=False, dropout=0.5))

model.add(Dense(num\_classes, activation='softmax'))

model.summary()

Layer (type) Output Shape Param #

=================================================================

time\_distributed\_10 (TimeDis (None, 10, 64, 64, 32) 2432

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_11 (TimeDis (None, 10, 62, 62, 32) 9248

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_12 (TimeDis (None, 10, 31, 31, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_4 (Dropout) (None, 10, 31, 31, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_13 (TimeDis (None, 10, 31, 31, 64) 18496

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_14 (TimeDis (None, 10, 31, 31, 64) 36928

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_15 (TimeDis (None, 10, 15, 15, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_5 (Dropout) (None, 10, 15, 15, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_16 (TimeDis (None, 10, 15, 15, 124) 71548

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_17 (TimeDis (None, 10, 15, 15, 124) 138508

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_18 (TimeDis (None, 10, 7, 7, 124) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_6 (Dropout) (None, 10, 7, 7, 124) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_19 (TimeDis (None, 10, 7, 7, 256) 285952

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_20 (TimeDis (None, 10, 7, 7, 256) 590080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_21 (TimeDis (None, 10, 3, 3, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_7 (Dropout) (None, 10, 3, 3, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_22 (TimeDis (None, 10, 3, 3, 512) 1180160

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_23 (TimeDis (None, 10, 3, 3, 512) 2359808

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_24 (TimeDis (None, 10, 1, 1, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_8 (Dropout) (None, 10, 1, 1, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_25 (TimeDis (None, 10, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_9 (Dropout) (None, 10, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

gru\_1 (GRU) (None, 256) 590592

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 5) 1285

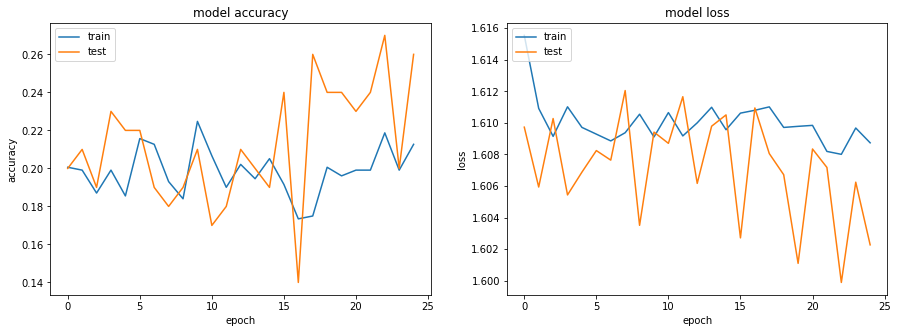
=================================================================

Total params: 5,285,037

Trainable params: 5,285,037

Non-trainable params: 0

Results



0\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*0

Experiments – 11 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code9

**Decision/ Explanation:**

num\_epochs = 50

remove drop after each layer

**Other Parameters from Previous Experiment**

kernel size. 3x3

Image Size = 128x128

Batch Size = 40

img\_idx = np.array( range(0,30,3)) #10 images

num\_epochs = 25 (reduced batch size)

optimiser = "adam"

steps\_per\_epoch = int(num\_train\_sequences/batch\_size) => 17

validation\_steps = int(num\_val\_sequences/batch\_size)  => 2

**Architecture**

#### Set1

model.add(TimeDistributed(Conv2D(32, (5, 5), strides=(2, 2), activation='relu',

padding='same'), input\_shape=(imgs\_from\_1video, imsize[0],imsize[1], 3)))

model.add(TimeDistributed(Conv2D(32, (3,3), kernel\_initializer="he\_normal", activation='relu')))

#model.add(BatchNormalization())

model.add(TimeDistributed( MaxPooling2D((2, 2), strides=(2, 2))) )

#model.add(Dropout(0.25))

#### Set2

model.add(TimeDistributed(Conv2D(64, (3,3), padding='same', activation='relu')))

model.add(TimeDistributed(Conv2D(64, (3,3), padding='same', activation='relu')))

#model.add(BatchNormalization())

model.add(TimeDistributed( MaxPooling2D(pool\_size=(2, 2))) )

#model.add(Dropout(0.25))

#### Set3

model.add(TimeDistributed(Conv2D(124, (3,3), padding='same', activation='relu')))

model.add(TimeDistributed(Conv2D(124, (3,3), padding='same', activation='relu')))

#model.add(BatchNormalization())

model.add(TimeDistributed( MaxPooling2D(pool\_size=(2, 2))) )

#model.add(Dropout(0.25))

#### Set4

model.add(TimeDistributed(Conv2D(256, (3,3), padding='same', activation='relu')))

model.add(TimeDistributed(Conv2D(256, (3,3), padding='same', activation='relu')))

#model.add(BatchNormalization())

model.add(TimeDistributed( MaxPooling2D(pool\_size=(2, 2))) )

model.add(Dropout(0.25))

#### Set3

model.add(TimeDistributed(Conv2D(512, (3,3), padding='same', activation='relu')))

model.add(TimeDistributed(Conv2D(512, (3,3), padding='same', activation='relu')))

#model.add(BatchNormalization())

model.add(TimeDistributed( MaxPooling2D(pool\_size=(2, 2))))

#model.add(Dropout(0.25))

model.add(TimeDistributed(Flatten()))

model.add(Dropout(0.5))

model.add(GRU(256, return\_sequences=False, dropout=0.5))

model.add(Dense(num\_classes, activation='softmax'))

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

time\_distributed\_1 (TimeDist (None, 10, 64, 64, 32) 2432

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_2 (TimeDist (None, 10, 62, 62, 32) 9248

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_3 (TimeDist (None, 10, 31, 31, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_4 (TimeDist (None, 10, 31, 31, 64) 18496

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_5 (TimeDist (None, 10, 31, 31, 64) 36928

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_6 (TimeDist (None, 10, 15, 15, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_7 (TimeDist (None, 10, 15, 15, 124) 71548

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_8 (TimeDist (None, 10, 15, 15, 124) 138508

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_9 (TimeDist (None, 10, 7, 7, 124) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_10 (TimeDis (None, 10, 7, 7, 256) 285952

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_11 (TimeDis (None, 10, 7, 7, 256) 590080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_12 (TimeDis (None, 10, 3, 3, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_1 (Dropout) (None, 10, 3, 3, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_13 (TimeDis (None, 10, 3, 3, 512) 1180160

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_14 (TimeDis (None, 10, 3, 3, 512) 2359808

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_15 (TimeDis (None, 10, 1, 1, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_16 (TimeDis (None, 10, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_2 (Dropout) (None, 10, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

gru\_1 (GRU) (None, 256) 590592

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 5) 1285

=================================================================

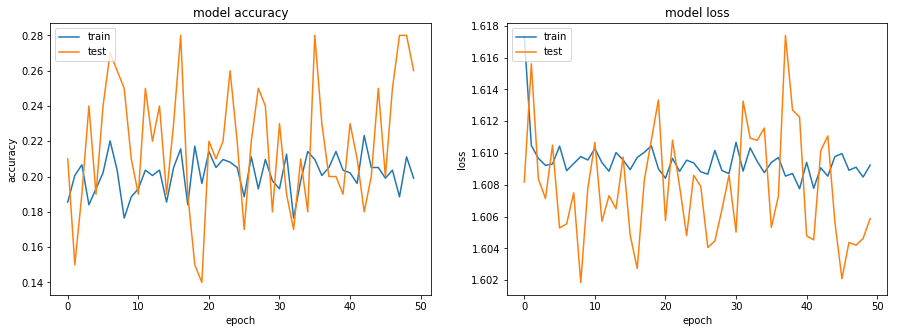
Total params: 5,285,037

Trainable params: 5,285,037

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

None



Experiments – 12 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code10

**Decision/ Explanation:**

Batch size=100

add batch normalization

**Other Parameters from Previous Experiment**

kernel size. 3x3

Image Size = 128x128

Batch Size = 40

num\_epochs = 50

img\_idx = np.array( range(0,30,3)) #10 images

num\_epochs = 25 (reduced batch size)

optimiser = "adam"

steps\_per\_epoch = int(num\_train\_sequences/batch\_size) => 17

validation\_steps = int(num\_val\_sequences/batch\_size)  => 2

#### Set1

model.add(TimeDistributed(Conv2D(32, (5, 5), strides=(2, 2), activation='relu',

padding='same'), input\_shape=(imgs\_from\_1video, imsize[0],imsize[1], 3)))

model.add(TimeDistributed(Conv2D(32, (3,3), kernel\_initializer="he\_normal", activation='relu')))

model.add(BatchNormalization())

model.add(TimeDistributed( MaxPooling2D((2, 2), strides=(2, 2))) )

#model.add(Dropout(0.25))

#### Set2

model.add(TimeDistributed(Conv2D(64, (3,3), padding='same', activation='relu')))

model.add(TimeDistributed(Conv2D(64, (3,3), padding='same', activation='relu')))

model.add(BatchNormalization())

model.add(TimeDistributed( MaxPooling2D(pool\_size=(2, 2))) )

#model.add(Dropout(0.25))

#### Set3

model.add(TimeDistributed(Conv2D(124, (3,3), padding='same', activation='relu')))

model.add(TimeDistributed(Conv2D(124, (3,3), padding='same', activation='relu')))

model.add(BatchNormalization())

model.add(TimeDistributed( MaxPooling2D(pool\_size=(2, 2))) )

#model.add(Dropout(0.25))

#### Set4

model.add(TimeDistributed(Conv2D(256, (3,3), padding='same', activation='relu')))

model.add(TimeDistributed(Conv2D(256, (3,3), padding='same', activation='relu')))

model.add(BatchNormalization())

model.add(TimeDistributed( MaxPooling2D(pool\_size=(2, 2))) )

model.add(Dropout(0.25))

#### Set3

model.add(TimeDistributed(Conv2D(512, (3,3), padding='same', activation='relu')))

model.add(TimeDistributed(Conv2D(512, (3,3), padding='same', activation='relu')))

model.add(BatchNormalization())

model.add(TimeDistributed( MaxPooling2D(pool\_size=(2, 2))))

#model.add(Dropout(0.25))

model.add(TimeDistributed(Flatten()))

model.add(Dropout(0.5))

model.add(GRU(256, return\_sequences=False, dropout=0.5))

model.add(Dense(num\_classes, activation='softmax'))

#write your model here

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

time\_distributed\_1 (TimeDist (None, 10, 64, 64, 32) 2432

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_2 (TimeDist (None, 10, 62, 62, 32) 9248

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_1 (Batch (None, 10, 62, 62, 32) 128

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_3 (TimeDist (None, 10, 31, 31, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_4 (TimeDist (None, 10, 31, 31, 64) 18496

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_5 (TimeDist (None, 10, 31, 31, 64) 36928

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_2 (Batch (None, 10, 31, 31, 64) 256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_6 (TimeDist (None, 10, 15, 15, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_7 (TimeDist (None, 10, 15, 15, 124) 71548

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_8 (TimeDist (None, 10, 15, 15, 124) 138508

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_3 (Batch (None, 10, 15, 15, 124) 496

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_9 (TimeDist (None, 10, 7, 7, 124) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_10 (TimeDis (None, 10, 7, 7, 256) 285952

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_11 (TimeDis (None, 10, 7, 7, 256) 590080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_4 (Batch (None, 10, 7, 7, 256) 1024

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_12 (TimeDis (None, 10, 3, 3, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_1 (Dropout) (None, 10, 3, 3, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_13 (TimeDis (None, 10, 3, 3, 512) 1180160

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_14 (TimeDis (None, 10, 3, 3, 512) 2359808

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_5 (Batch (None, 10, 3, 3, 512) 2048

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_15 (TimeDis (None, 10, 1, 1, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

time\_distributed\_16 (TimeDis (None, 10, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_2 (Dropout) (None, 10, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

gru\_1 (GRU) (None, 256) 590592

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 5) 1285

=================================================================

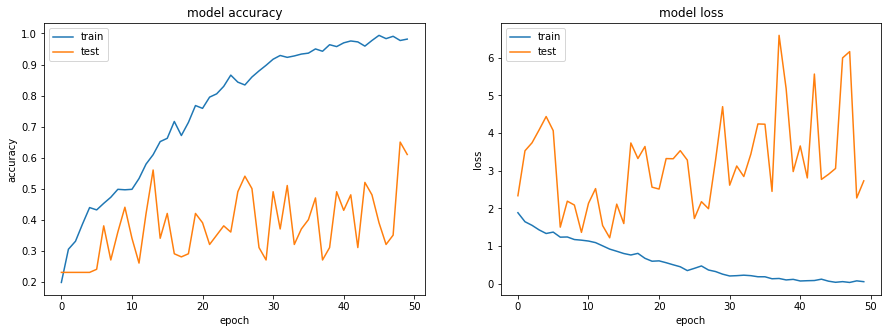
Total params: 5,288,989

Trainable params: 5,287,013

Non-trainable params: 1,976

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

None



Experiments – 13 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code11

**Decision/ Explanation:**

Batch size=100

num\_epochs = 100

Add dropout

Change kernel size for first layer Conv2D(32, (3, 3) (earlier 5,5)

**Other Parameters from Previous Experiment**

kernel size. 3x3

Image Size = 128x128

Batch Size = 40

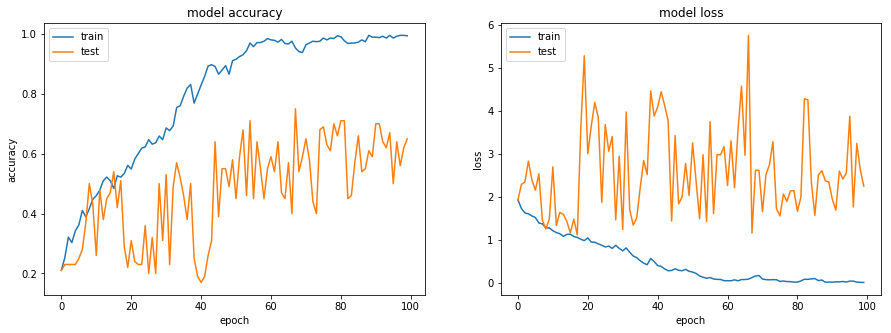
img\_idx = np.array( range(0,30,3)) #10 images

num\_epochs = 25 (reduced batch size)

optimiser = "adam"

steps\_per\_epoch = int(num\_train\_sequences/batch\_size) => 17

validation\_steps = int(num\_val\_sequences/batch\_size)  => 2



Experiments – 14 : CNN+RNN

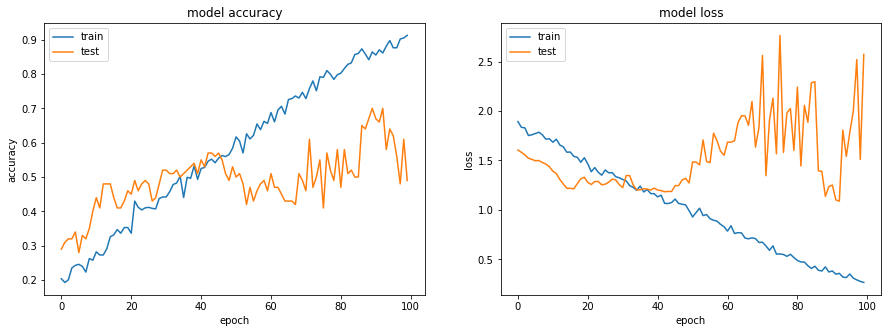
jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code12

**Decision/ Explanation:**

* optimiser = Adam(lr=0.0001) #write your optimizer

**Other Parameters from Previous Experiment**

* kernel size. 3x3
* Image Size = 128x128
* Batch size=100
* num\_epochs = 100
* img\_idx = np.array( range(0,30,3)) #10 images
* steps\_per\_epoch = int(num\_train\_sequences/batch\_size) => 17
* validation\_steps = int(num\_val\_sequences/batch\_size)  => 2



Experiments – 13 : CNN+RNN

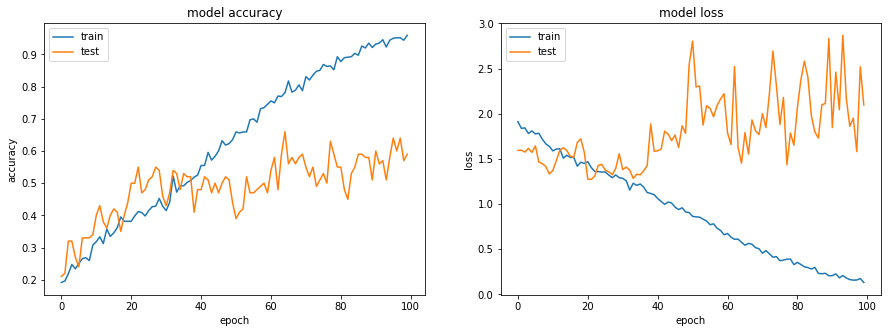
jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code13

**Decision/ Explanation:**

* kernel size changed = (4,4) earlier (3,3)

**Other Parameters from Previous Experiment**

* kernel size. 4x4
* Image Size = 128x128
* Batch size=100
* num\_epochs = 100
* img\_idx = np.array( range(0,30,3)) #10 images
* steps\_per\_epoch = int(num\_train\_sequences/batch\_size) => 17
* validation\_steps = int(num\_val\_sequences/batch\_size)  => 2



Experiments – 14 : CNN+RNN

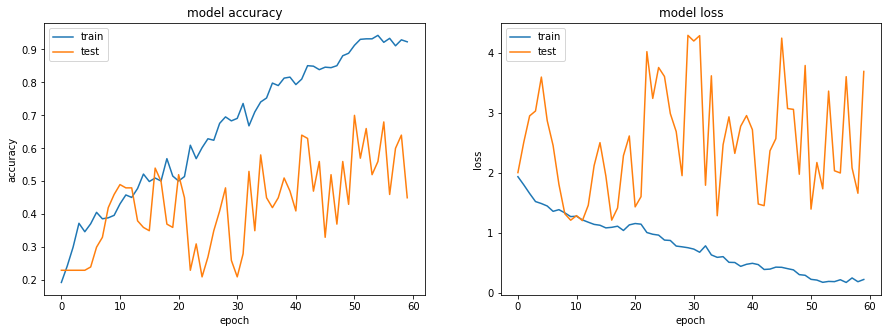
jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code14

**Decision/ Explanation:**

* kernel size changed = (5,5) earlier (4,4)
* optimiser = Adam(lr=0.0005)
* Num\_epochs = 60

**Other Parameters from Previous Experiment**

* kernel size. 4x4
* Image Size = 128x128
* Batch size=100
* img\_idx = np.array( range(0,30,3)) #10 images
* steps\_per\_epoch = int(num\_train\_sequences/batch\_size) => 17

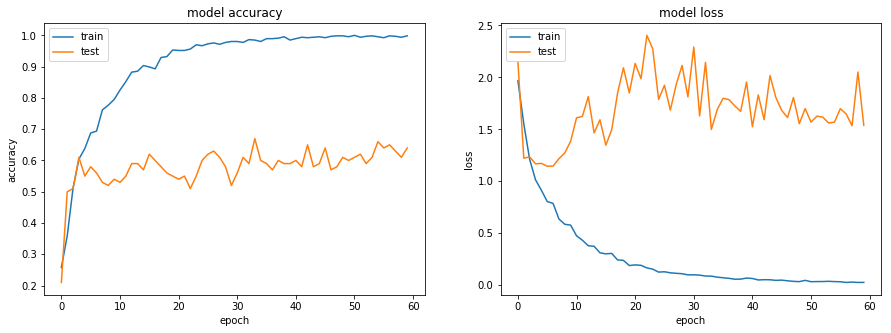


Experiments – 15 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code15

**Decision/ Explanation:**

* kernel size changed = (3,3)
* optimiser = Adam(lr=0.0005)
* Num\_epochs = 60
* Only one layer (all removed)
* Dropout .80

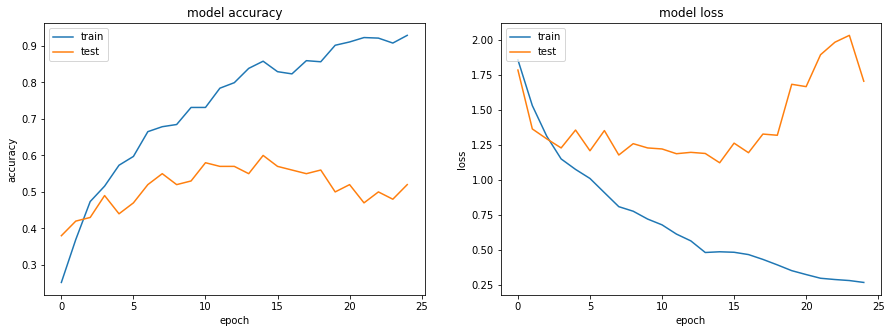


Experiments – 16 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code16

**Decision/ Explanation:**

* kernel size changed = (5,5) earlier (4,4)
* optimiser = Adam(lr=0.0005)
* Num\_epochs = 25
* GRU 128
* Dropout .80

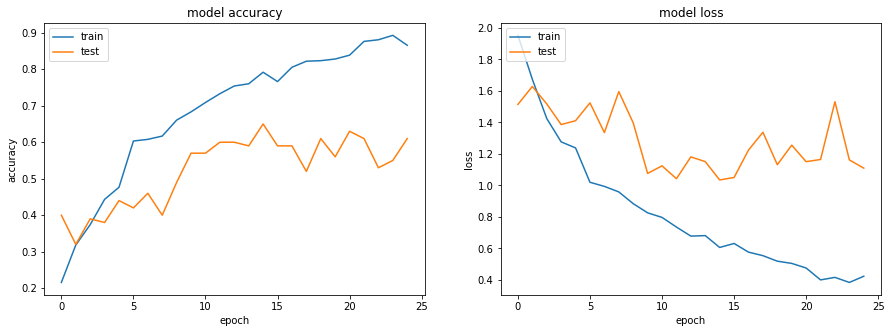


Experiments – 17 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code16 (same as before)

**Decision/ Explanation:**

* kernel size changed = (5,5) earlier (4,4)
* optimiser = Adam(lr=0.0005)
* Num\_epochs = 25
* GRU 64
* Dropout .80

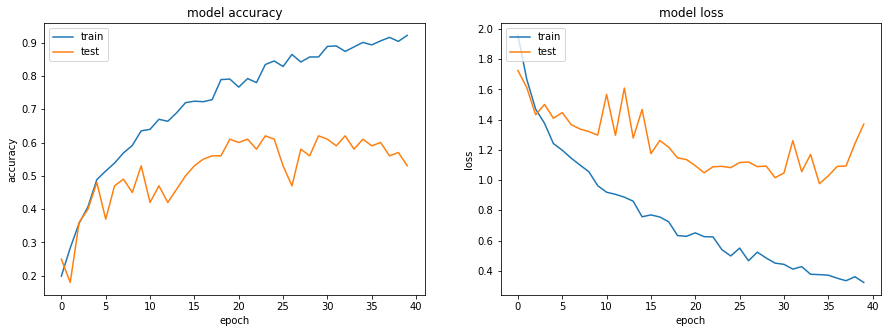


Experiments – 18 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code16 (same as before)

**Decision/ Explanation:**

* kernel size changed = (5,5) earlier (4,4)
* optimiser = Adam(lr=0.0005)
* Num\_epochs = 60
* GRU 32
* Dropout .80

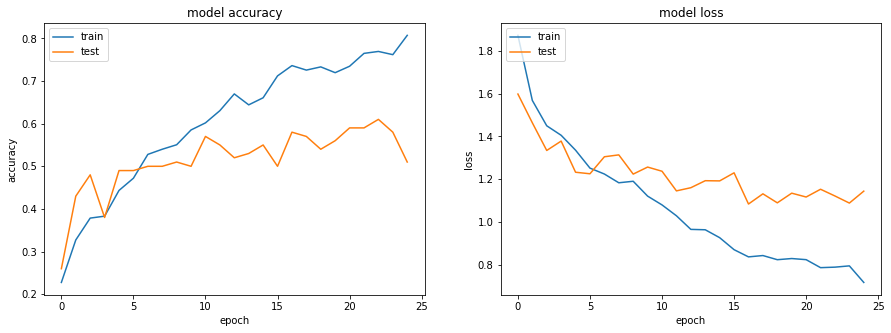


Experiments – 19 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code16 (same as before)

**Decision/ Explanation:**

* Only one CONV2D layer (all other removed).
* Conv2D(90)

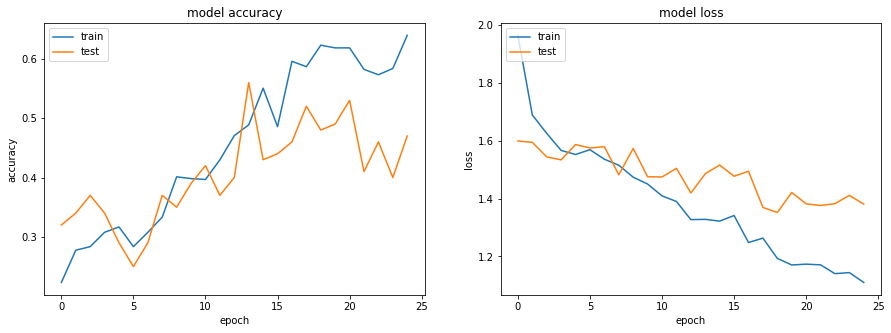


Experiments – 20 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code16 (same as before)

**Decision/ Explanation:**

* drop from GRU layer neurons. model.add(GRU(16, return\_sequences=False))

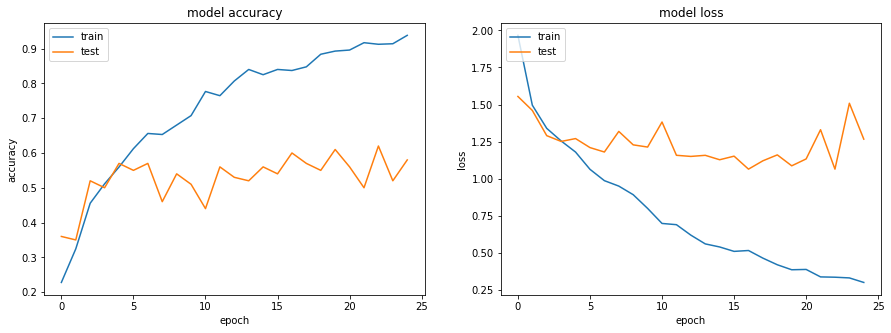


Experiments – 21 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code16 (same as before)

**Decision/ Explanation:**

* drop from GRU layer neurons. model.add(GRU(64, return\_sequences=False))

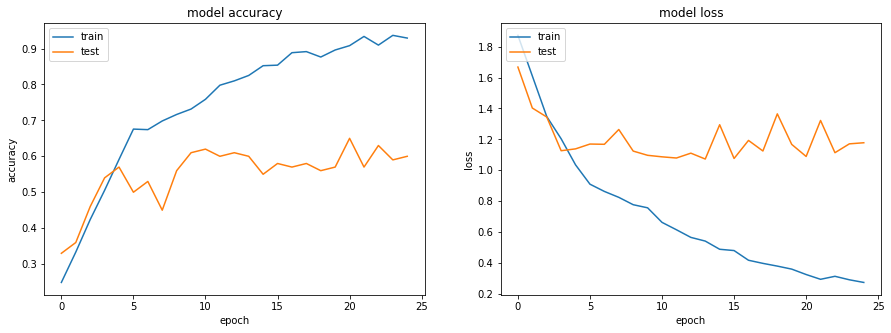


Experiments – 22 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code16 (same as before)

**Decision/ Explanation:**

* Conv2D(45, (3,3)

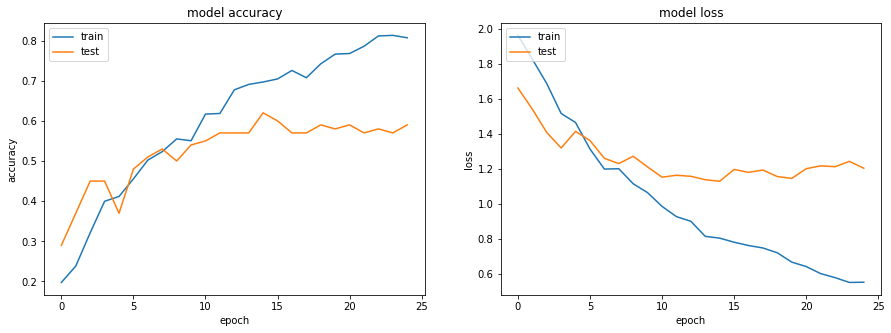


Experiments – 23 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code16 (same as before)

**Decision/ Explanation:**

* Conv2D(15, (3,3)

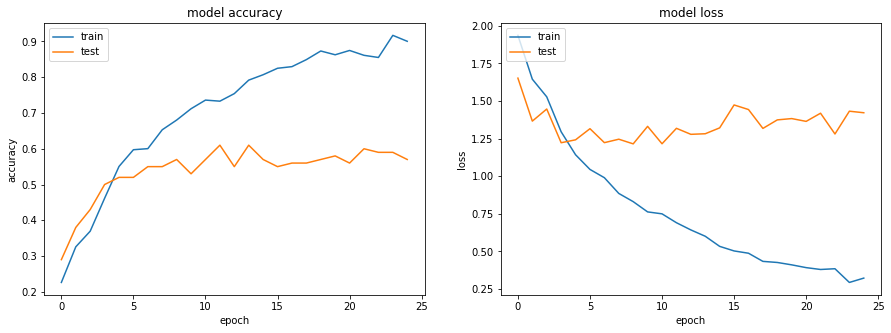


Experiments – 24 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code16 (same as before)

**Decision/ Explanation:**

* Conv2D(20, (3,3)
* GRU 128

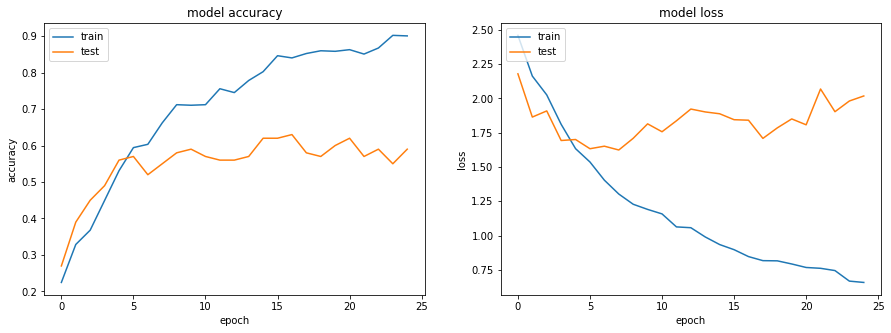


Experiments – 25 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code16 (same as before)

**Decision/ Explanation:**

* kernel\_regularizer=l2(0.1)

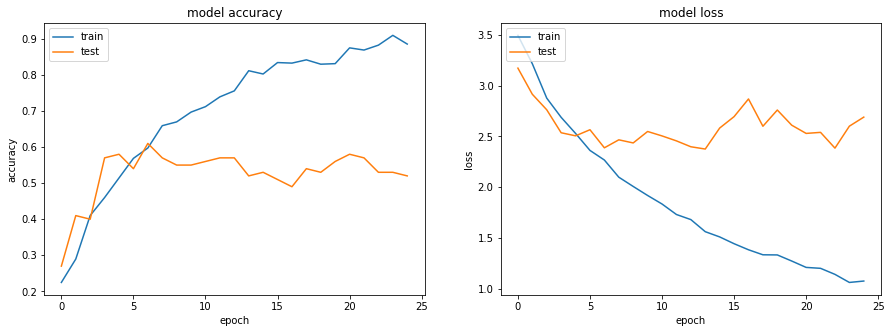


Experiments – 26 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code16 (same as before)

**Decision/ Explanation:**

* kernel\_regularizer=l2(0.3)

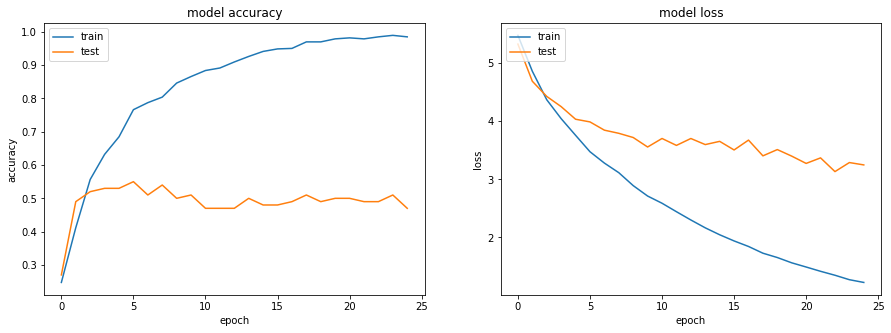


Experiments – 27 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code16 (same as before)

**Decision/ Explanation:**

* kernel\_regularizer=l2(0.7)
* model.add(GRU(128, return\_sequences=False)) (dropout removed)

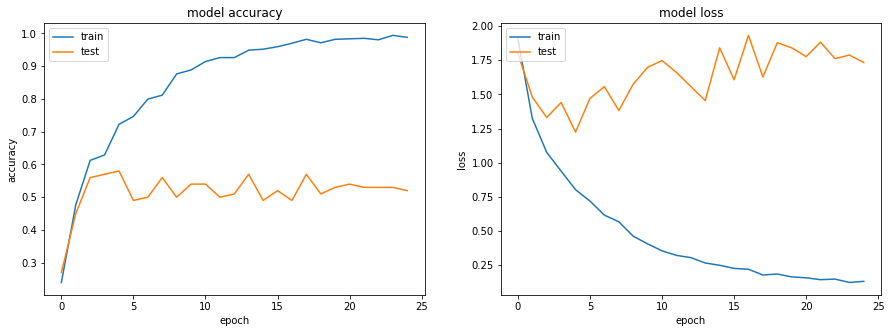


Experiments – 28 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code16 (same as before)

**Decision/ Explanation:**

* kernel\_regularizer=l2(0.001)
* model.add(GRU(128, return\_sequences=False)) (dropout removed)

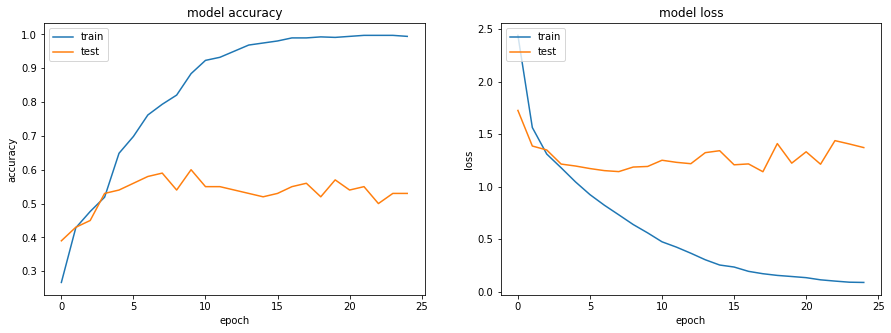


Experiments – 29 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code16 (same as before)

**Decision/ Explanation:**

* Removed regularization
* Kept three layers each of 32 neuron and kernel 3x3
* model.add(Dropout(0.5))
* GRU(64)

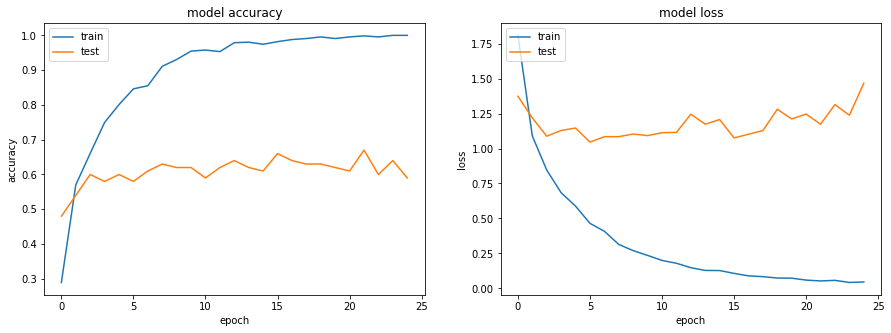


Experiments – 30 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code16 (same as before)

**Decision/ Explanation:**

* Removed regularization
* Kept three layers each of 32 neuron and kernel 3x3
* model.add(Dropout(0.5))
* GRU(64)



Resetting the Entire Architecture.

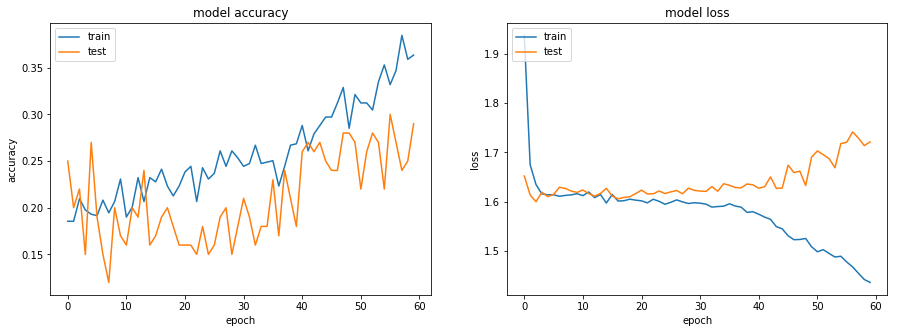
**Accuracy not improving. Changing Strategy and recreating architecture.**

Experiments – 31 : CNN+RNN

Jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code17

**Decision/ Explanation:**

* Epoch 60
* Conv2D(64, (3,3), strides=(2, 2), activation='relu', padding='same', kernel\_initializer="he\_normal")
* model.add(BatchNormalization())
* model.add(TimeDistributed( MaxPooling2D((3, 3), strides=(2, 2))) )
* model.add(TimeDistributed(Flatten()))
* model.add(GRU(32, return\_sequences=False))

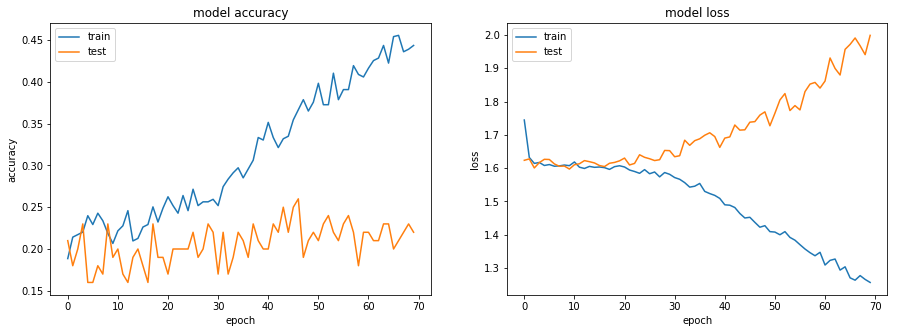


Experiments – 32 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code17 (same as before)

**Decision/ Explanation:**

* Add one more Conv2D Layer
* Increase epoch to 70

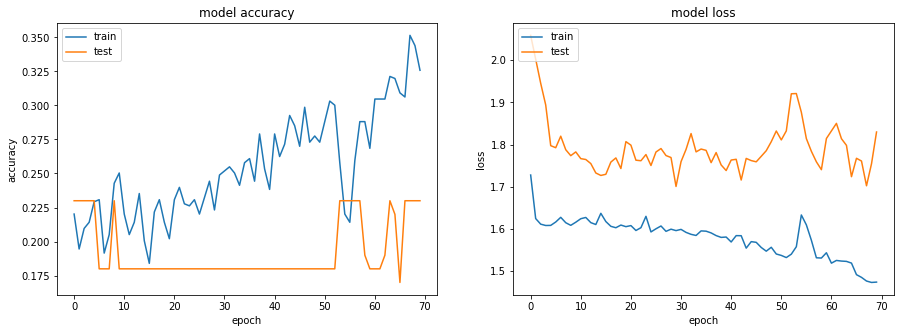


Experiments – 33 : CNN+RNN

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code17 (same as before)

**Decision/ Explanation:**

* Because there is huge difference between Train and Val accuracy so adding dropout & normalization.
* At the end of conv2d (3rd layer)
  + model.add(BatchNormalization())
  + model.add(Dropout(0.50))
* Because ever after 70 epochs accuracy is very less, so Add one more Conv2D Layer (now 3 conv2d layers)

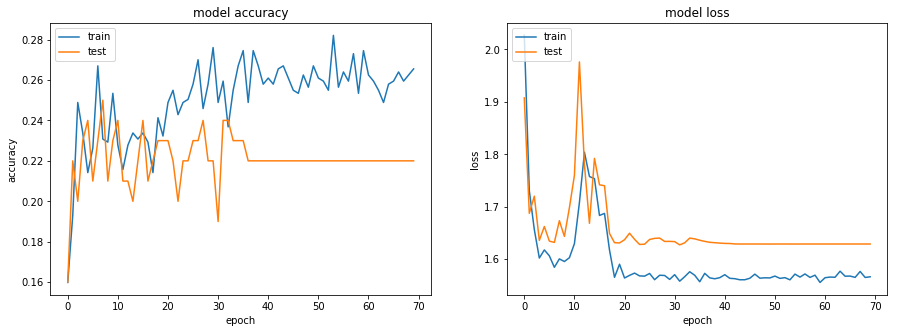


Experiments – 34 : CNN+RNN

**Jupyter\_file name:** Neural\_Nets\_Project\_Starter\_Code17 (same as before)

**Decision/ Explanation:**

* Because even basic learning is not happening so
  + Remove BatchNormalization
  + Remove Dropout
* Because model is learning slow. Even after 70 iteration no much gain, so
  + optimiser = Adam(lr=0.1) earlier .001
  + LR = ReduceLROnPlateau(monitor='val\_loss', factor=0.1, patience=10, verbose=0, mode='auto', min\_delta=0.001, cooldown=0, min\_lr=0)

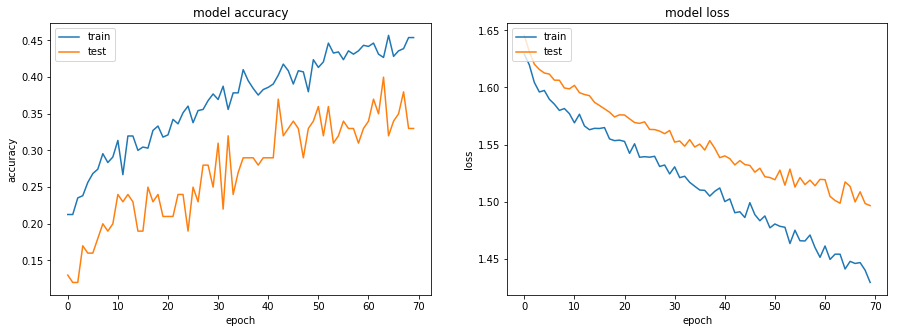


Experiments – 35 : CNN+RNN

**Jupyter\_file name:** Neural\_Nets\_Project\_Starter\_Code17 (same as before)

**Decision/ Explanation:**

* Because accuracy dropped so
  + Resetting lr to .001
  + Changing optimizer to SGD (earlier Adam)

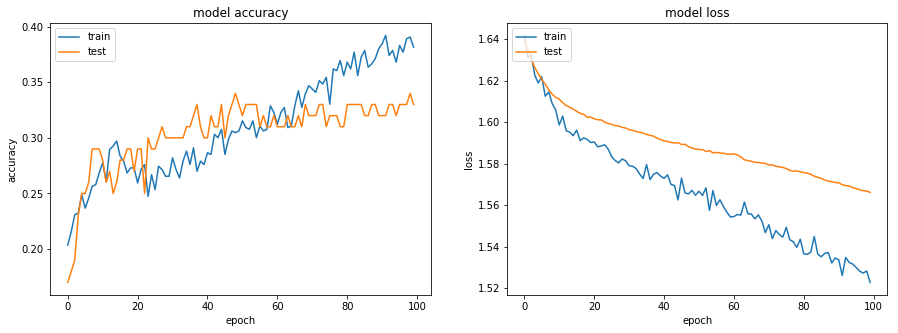


Experiments – 36 : CNN+RNN

**Jupyter\_file name:** Neural\_Nets\_Project\_Starter\_Code17 (same as before)

**Decision/ Explanation:**

* Training and Val accuracy continuous and simultaneously improving. So, increasing number of epoch to 100.
* Changing number of neurons on each Conv2D layer to 32. Earlier 64.

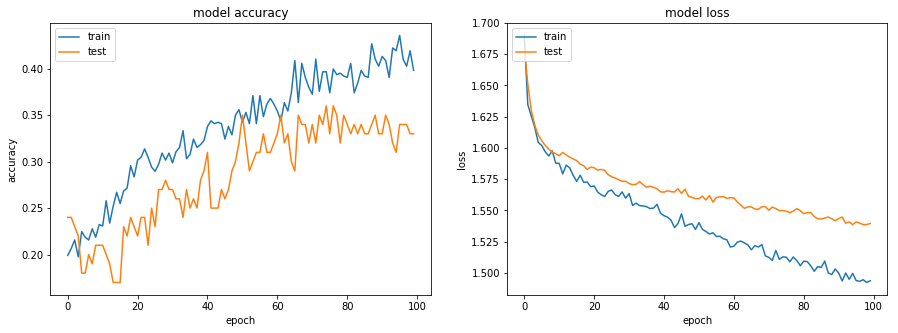


Experiments – 37 : CNN+RNN

**Jupyter\_file name:** Neural\_Nets\_Project\_Starter\_Code17 (same as before)

**Decision/ Explanation:**

* Conv2D (64) + Conv2D (64)+ MaxPool2D + Conv2D(128) + Conv2D(128) + MaxPool2D



**Accuracy not improving. Changing Strategy and recreating architecture.**

Experiments – 38 : CNN+RNN

model.add(TimeDistributed(Conv2D(32, (3,3), strides=(2, 2), activation='relu',

padding='same'), input\_shape=(imgs\_from\_1video, imsize[0],imsize[1], 3)))

#model.add(Dropout(0.25))

model.add(BatchNormalization())

model.add(TimeDistributed(Conv2D(32, (3,3), kernel\_initializer="he\_normal", activation='relu')))

#model.add(Dropout(0.25))

model.add(BatchNormalization())

model.add(TimeDistributed(Conv2D(32, (3,3), kernel\_initializer="he\_normal", activation='relu')))

model.add(Dropout(0.25))

model.add(BatchNormalization())

model.add(TimeDistributed( MaxPooling2D((3, 3), strides=(2, 2))) )

model.add(BatchNormalization())

model.add(TimeDistributed(Flatten()))

model.add(Dropout(0.5))

model.add(GRU(64, return\_sequences=False))

model.add(Dense(num\_classes, activation='softmax'))

######################################################################

Epoch =50

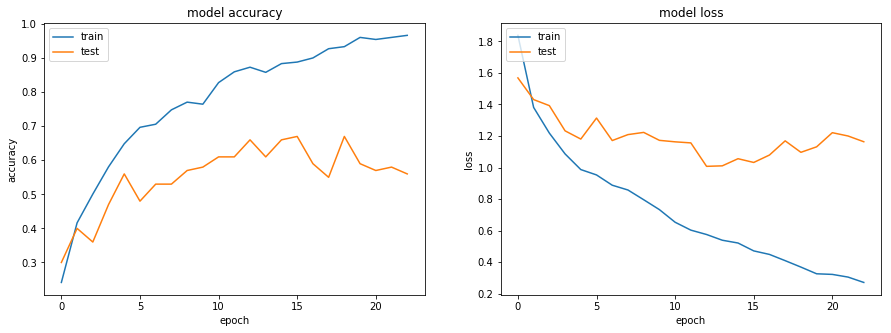
Conv2D(32(3,3)) + BatchNormalization +

Conv2D(32(3,3)) + BatchNormalization +

Conv2D(32(3,3)) + BatchNormalization +

MaxPooling2D((3, 3) strides=(2, 2) + BatchNormalization +

Flatten + dropout(.5) + GRU(64)



Experiments – 39 : CNN+RNN

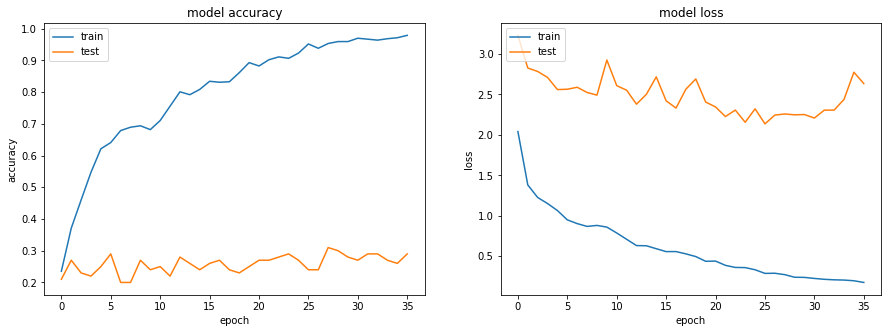
Conv2D(32(3,3)) + dropout(.25) + BatchNormalization +

Conv2D(32(3,3)) + dropout(.25) + BatchNormalization +

Conv2D(32(3,3)) + dropout(.25) + BatchNormalization +

MaxPooling2D((3, 3) strides=(2, 2) + BatchNormalization +

Flatten + dropout(.5) + GRU(64)



Experiments – 40 : CNN+RNN

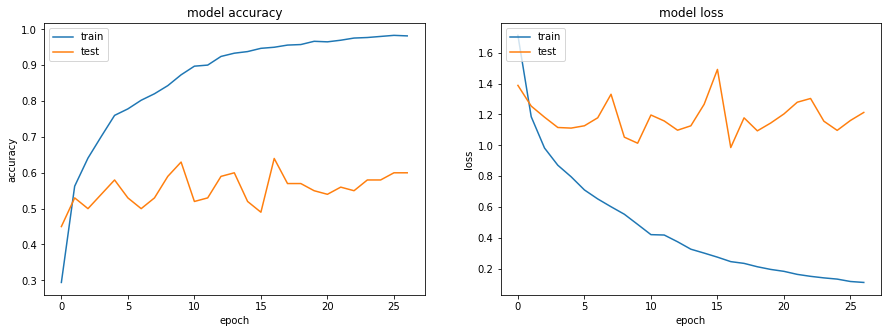
Conv2D(32(3,3)) + BatchNormalization +

Conv2D(32(3,3)) + BatchNormalization +

Conv2D(32(3,3)) + dropout(.25) + BatchNormalization +

MaxPooling2D((3, 3) strides=(2, 2) + BatchNormalization +

Flatten + dropout(.5) + GRU(64)



Experiments – 42 : CNN+RNN

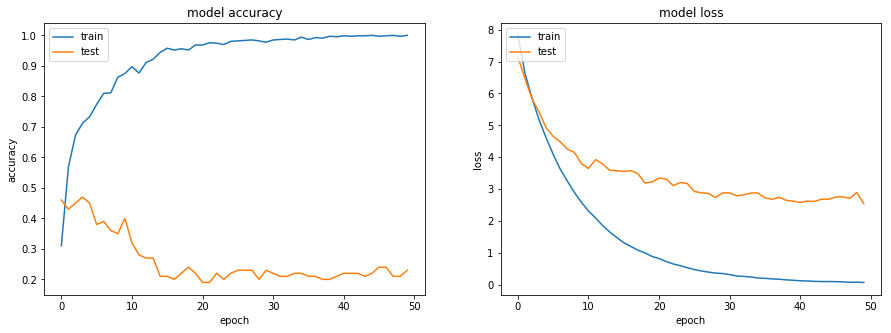
Conv2D(32(3,3)) + BatchNormalization +

Conv2D(32(3,3)) + BatchNormalization +

Conv2D(32(3,3)) + dropout(.25) + kernel\_regularizer=l2(.1)+ BatchNormalization +

MaxPooling2D((3, 3) strides=(2, 2) + BatchNormalization +

Flatten + dropout(.5) + GRU(64)



Experiments – 41 : CNN+RNN

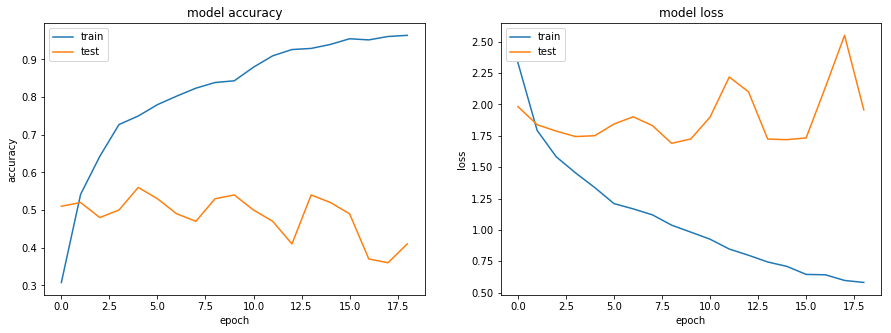
Conv2D(32(3,3)) + BatchNormalization +

Conv2D(32(3,3)) + BatchNormalization +

Conv2D(32(3,3)) + dropout(.25) + kernel\_regularizer=l2(.01)+ BatchNormalization +

MaxPooling2D((3, 3) strides=(2, 2) + BatchNormalization +

Flatten + dropout(.5) + GRU(64)



Experiments – 43 : CNN+RNN

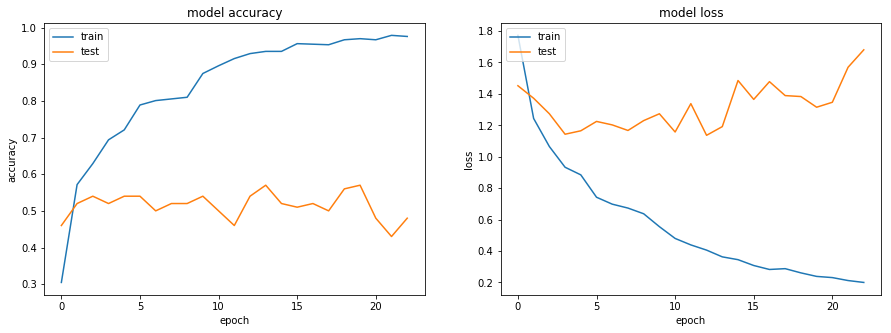
Conv2D(32(3,3)) + BatchNormalization +

Conv2D(32(3,3)) + BatchNormalization +

Conv2D(32(3,3)) + dropout(.25) + kernel\_regularizer=l2(.001)+ BatchNormalization +

MaxPooling2D((3, 3) strides=(2, 2) + BatchNormalization +

Flatten + dropout(.5) + GRU(64)



Experiments – 44 : CNN+RNN

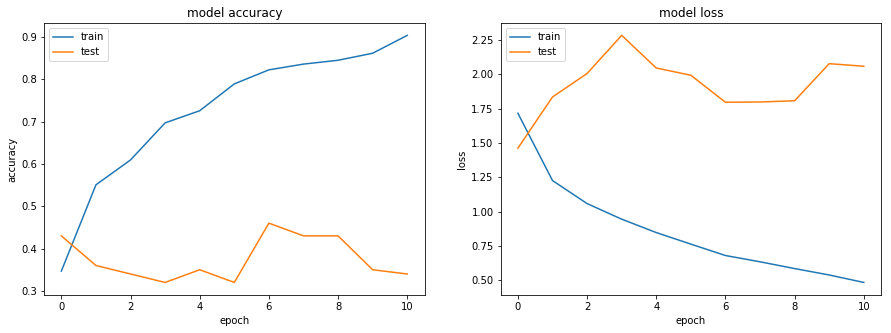
Conv2D(32(3,3)) + BatchNormalization +

Conv2D(32(3,3)) + BatchNormalization +

Conv2D(32(3,3)) + dropout(.5) + kernel\_regularizer=l2(.001)+ BatchNormalization +

MaxPooling2D((3, 3) strides=(2, 2) + BatchNormalization +

Flatten + dropout(.5) + GRU(64)



Experiments – 45 : CNN+RNN

Conv2D(32(3,3)) + BatchNormalization +

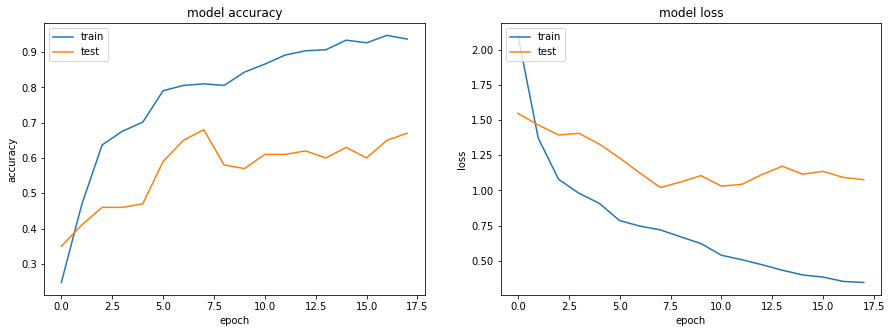
Conv2D(32(3,3)) + BatchNormalization +

Conv2D(32(3,3)) + BatchNormalization +

Conv2D(32(3,3)) + dropout(.5) + kernel\_regularizer=l2(.001)+ BatchNormalization +

MaxPooling2D((3, 3) strides=(2, 2) + BatchNormalization +

Flatten + dropout(.5) + GRU(64)



Experiments – 46 : CNN+RNN

Conv2D(32(3,3)) + BatchNormalization +

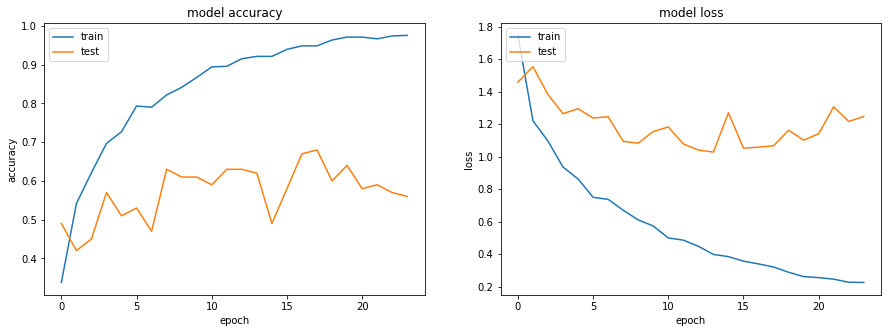
Conv2D(32(3,3)) + BatchNormalization +

Conv2D(32(3,3)) + BatchNormalization +

Conv2D(32(3,3)) + kernel\_regularizer=l2(.001)+ BatchNormalization +

MaxPooling2D((3, 3) strides=(2, 2) + BatchNormalization +

Flatten + dropout(.5) + GRU(64)



Experiments – 47 : CNN+RNN

Conv2D(32(3,3)) + BatchNormalization +

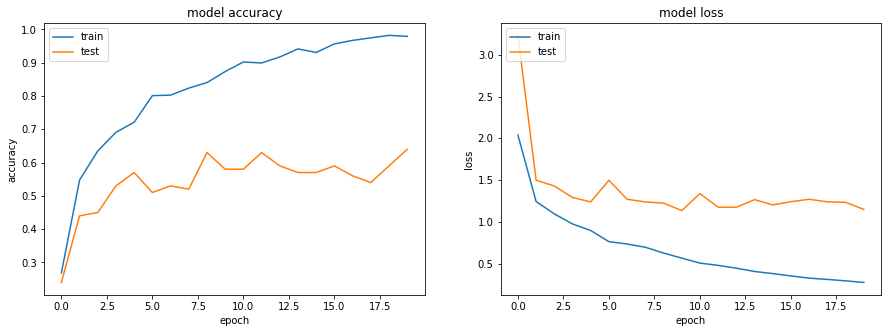
Conv2D(32(3,3)) + BatchNormalization +

Conv2D(32(3,3)) + BatchNormalization + kernel\_regularizer=l2(.001)+

Conv2D(32(3,3)) + kernel\_regularizer=l2(.001)+ BatchNormalization +

MaxPooling2D((3, 3) strides=(2, 2) + BatchNormalization +

Flatten + dropout(.5) + GRU(64)



Experiments – 48 : CNN+RNN

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

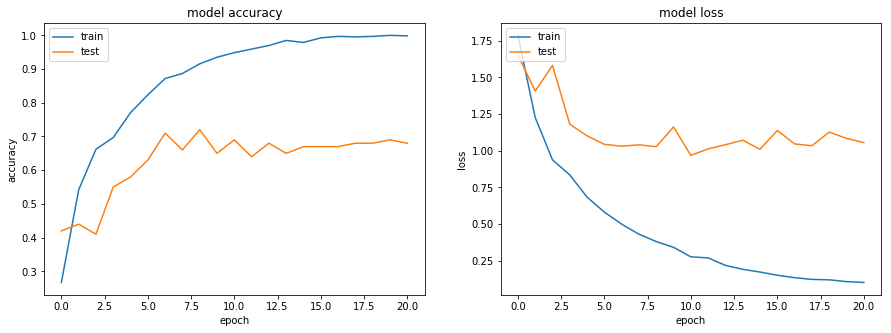
MaxPooling2D((3, 3) strides=(2, 2)

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

Conv2D(32(3,3)) strides=(2, 2) + kernel\_regularizer=l2(.001)+ BatchNormalization +

MaxPooling2D((3, 3) strides=(2, 2) + BatchNormalization +

Flatten + dropout(.5) + GRU(64)



Experiments – 49 : CNN+RNN

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

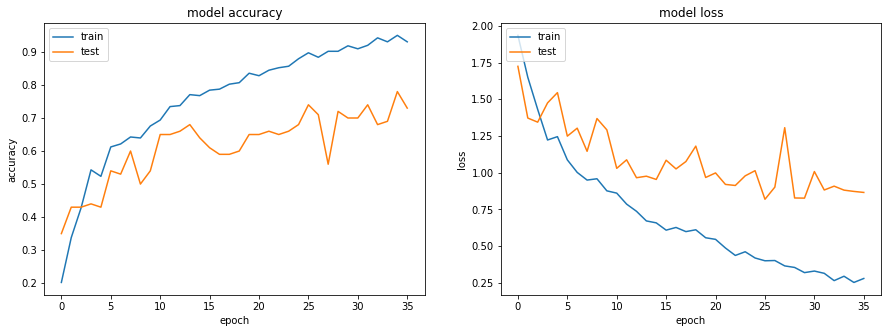
MaxPooling2D((3, 3) strides=(2, 2)

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

Conv2D(32(3,3)) strides=(2, 2) + kernel\_regularizer=l2(.001)+ BatchNormalization +

MaxPooling2D((3, 3) strides=(2, 2) + BatchNormalization +

Flatten + dropout(.8) + GRU(64)



Experiments – 50 : CNN+RNN

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

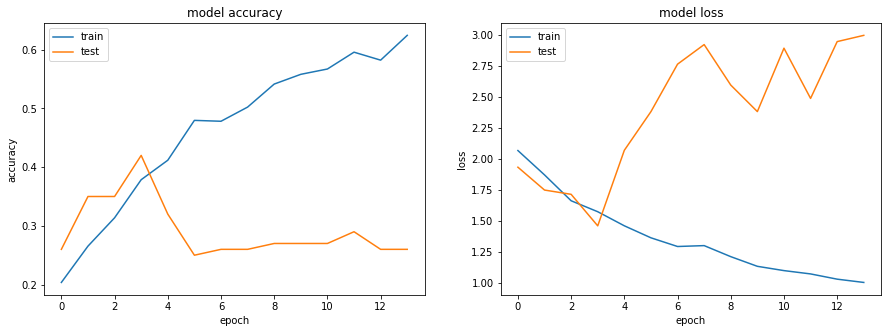
MaxPooling2D((3, 3) strides=(2, 2)

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

Conv2D(32(3,3)) strides=(2, 2) + kernel\_regularizer=l2(.001)+ BatchNormalization +

MaxPooling2D((3, 3) strides=(2, 2) + BatchNormalization +

Flatten + dropout(.9) + GRU(64)



Experiments – 51 : CNN+RNN

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

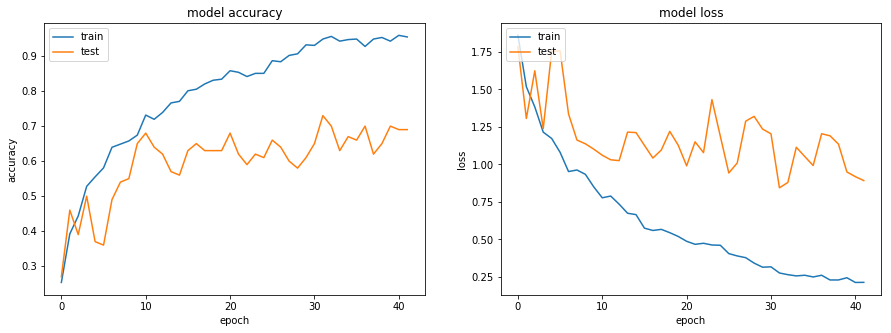
MaxPooling2D((3, 3) strides=(2, 2)

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

Conv2D(32(3,3)) strides=(2, 2) + kernel\_regularizer=l2(.001)+ BatchNormalization +

MaxPooling2D((3, 3) strides=(2, 2) + BatchNormalization +

Flatten + dropout(.8) + GRU(64)



Experiments – 52 : CNN+RNN

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

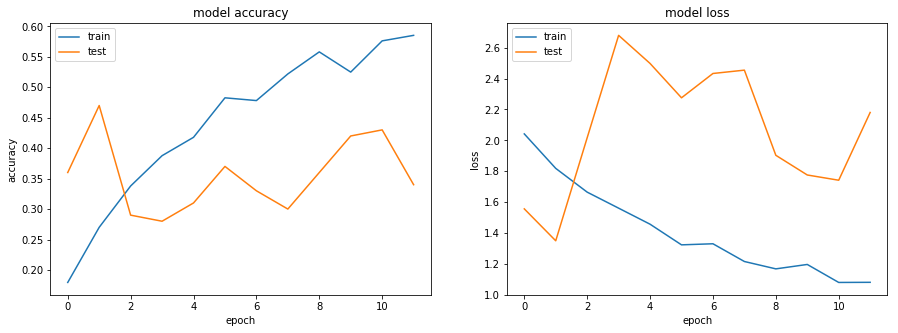
MaxPooling2D((3, 3) strides=(2, 2)

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

Conv2D(32(3,3)) strides=(2, 2) + kernel\_regularizer=l2(.001)+ BatchNormalization +

MaxPooling2D((3, 3) strides=(2, 2) + BatchNormalization +

Flatten + dropout(.9) + GRU(64)



Experiments – 53 : CNN+RNN

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

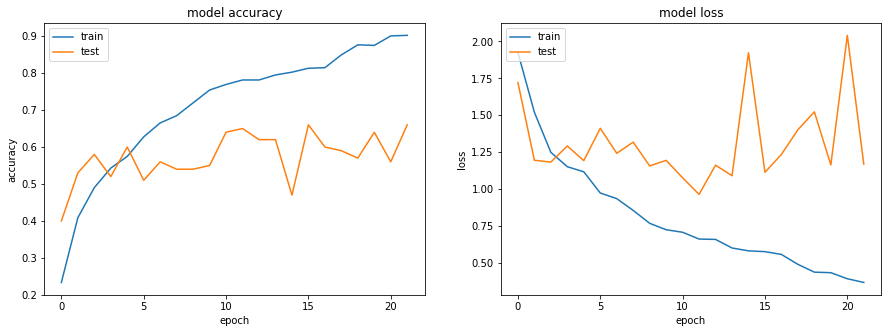
MaxPooling2D((3, 3) strides=(2, 2)

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

Conv2D(32(3,3)) strides=(2, 2) + kernel\_regularizer=l2(.001)+ BatchNormalization +

MaxPooling2D((3, 3) strides=(2, 2) + BatchNormalization +

Flatten + dropout(.8) + GRU(84)



Experiments – 54 : CNN+RNN

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

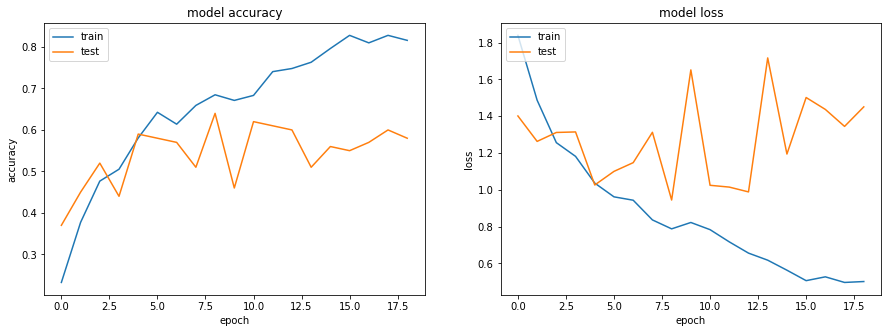
MaxPooling2D((3, 3) strides=(2, 2)

Conv2D(32(3,3)) strides=(2, 2) + BatchNormalization +

Conv2D(32(3,3)) strides=(2, 2) + kernel\_regularizer=l2(.001)+ BatchNormalization +

MaxPooling2D((3, 3) strides=(2, 2) + BatchNormalization +

Flatten + dropout(.8) + GRU(48)



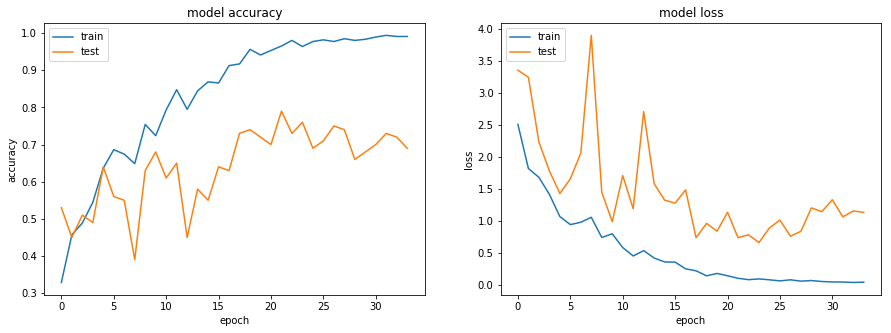
**Conclusion:**

**CNN: Experiments – No : 9**

jupyter\_file name: Neural\_Nets\_Project\_Starter\_Code5

**Model: model-00024-0.09463-0.96380-0.66293-0.76000.h5**

Validation Accuracy: 76%, Training Accuracy: 95%



**CNN+RNN: Experiments – No: 49**

Neural\_Nets\_Project\_Starter\_Code18

**Model: CNN\_RNN-model-00035-0.16339-0.95324-0.82417-0.72000**

Validation Accuracy: 72%, Training Accuracy: 95%

