|  |  |  |  |
| --- | --- | --- | --- |
| Use LSTM , epochs = 10, batchsize = 100 |  | Each epoch is taking over 20 mins to train on google colab. | Too high training time. Try smaller batch size |
| Use LSTM , epochs = 10, batchsize = 10 |  | loss: 0.3958 - categorical\_accuracy: 0.8537 - val\_loss: 4.9297 - val\_categorical\_accuracy: 0.2100 | Model is overfitting |
| Use GRUu , epochs = 10, batchsize = 10 |  | Cat\_loss came out to nan |  |
| Remove dropouts from 2 and 3rd convolution layer |  | Cat\_loss came out to nan |  |
|  |  |  |  |
| Add affine transformation to data, 5 epocs, 100 videos only. |  | loss: 0.8562  categorical\_accuracy: 0.7267  val\_loss: 3.1973  val\_categorical\_accuracy: 0.4800 |  |
| Add affine transformation to data,  100 videos only ,  10 epochs  Size = 120 |  | loss: 0.7928  categorical\_accuracy: 0.7267  val\_loss: 3.3824  val\_categorical\_accuracy: 0.4500 |  |
| Add affine transformation to data,  10 epocs,  100 videos.  Size reduced to 84 |  | Model is still over fitting but  loss: 0.9580 - categorical\_accuracy: 0.6967 - val\_loss: 2.6072 - val\_categorical\_accuracy: 0.4200 |  |
| 10 epochs  100 videos  Size = 84\*84  Increase batch size = 30 |  | loss: 0.7743  categorical\_accuracy: 0.7250  val\_loss: 1.8595  val\_categorical\_accuracy: 0.3500 |  |
| Epochs = 10  Videos = 663  Size= 84\*84  Batch size = 30 |  | Model is drastically overfitting  loss: 1.7697  categorical\_accuracy: 0.4396  val\_loss: 2.0745  val\_categorical\_accuracy: 0.2000 |  |
| Epochs = 10  Videos = 663  Size= 120\*120  Batch size = 30 | nimblebox | loss: 1.6568  categorical\_accuracy: 0.4300  val\_loss: 2.0831  val\_categorical\_accuracy: 0.5250 |  |
| Dense layer = 64, 32  Epochs = 10  Videos = 100  Size= 120\*120  Batch size = 10 | colab | loss: 0.8811  categorical\_accuracy: 0.8300  val\_loss: 5.1926  val\_categorical\_accuracy: 0.3700 |  |
| Videos = 663  Size = 120\*120  Epochs = 10  Batch size = 30 |  | loss: 2.3570  categorical\_accuracy: 0.3478  val\_loss: 6.6927  val\_categorical\_accuracy: 0.2500 |  |
|  |  |  |  |
|  |  |  |  |
| Use LSTM, epochs = 10, ReduceLROnPlateau( monitor="val\_loss", factor=0.1, patience=10, verbose=0, mode="auto", min\_delta=0.0001, cooldown=0, min\_lr=0, \*\*kwargs) |  |  |  |