**Data Centre Vision – UCS665**

**Evaluation 2**

|  |  |
| --- | --- |
| **Submitted by:** | **Submitted to:**  **Dr. Sachin Kansal** |
| **Samprad Raja Uprety (101917094)** |
| **Anshul Tungaria (101917091)** |
| **Apoorv Taneja (102097006)** |

**BE Third Year- CSE**



**Computer Science and Engineering Department**

**Thapar Institute of Engineering and Technology, Patiala**

**March, 2022**

**Parameter Table:**

|  |  |
| --- | --- |
| **Parameters** | **Values** |
| Model | ConvLSTM |
| Train, Validation and Test Ratio | Ratio = (0.75, 0, 0.25) |
| Optimization Algorithm | Adam Optimizer |
| Learning Rate in Adam Optimizer | Learning rate = 0.001 |
| Activation Function | softmax |
| Loss Function | loss='categorical\_crossentropy' |
| Number of Hidden layers | 10 |
| Number of iterations(epochs) | 100 |
| Batch Size | 10 |
| Kernel size | (3,3) |

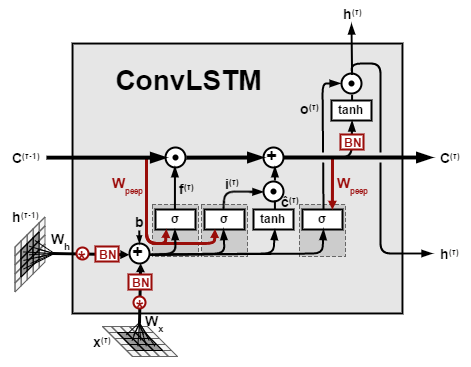
**ConvLSTM**

Data collected over successive periods of time are characterised as a Time Series. In such cases, an interesting approach is to use a model based on LSTM (Long Short-Term Memory), a Recurrent Neural Network architecture. In this kind of architecture, the model passes the previous hidden state to the next step of the sequence. Therefore, holding information on previous data, the network has seen before and using it to make decisions. In other words, the data order is extremely important.

When working with images, the best approach is a CNN (Convolutional Neural Network) architecture. The image passes through Convolutional Layers, in which several filters extract important features. After passing some convolutional layers in sequence, the output is connected to a fully-connected Dense network.

In our case, sequencial images, one approach is using ConvLSTM layers. It is a Recurrent layer, just like the LSTM, but internal matrix multiplications are exchanged with convolution operations. As a result, the data that flows through the ConvLSTM cells keeps the input dimension (3D in our case) instead of being just a 1D vector with features.

A different approach of a ConvLSTM is a Convolutional-LSTM model, in which the image passes through the convolutions layers and its result is a set flattened to a 1D array with the obtained features. When repeating this process to all images in the time set, the result is a set of features over time, and this is the LSTM layer input.

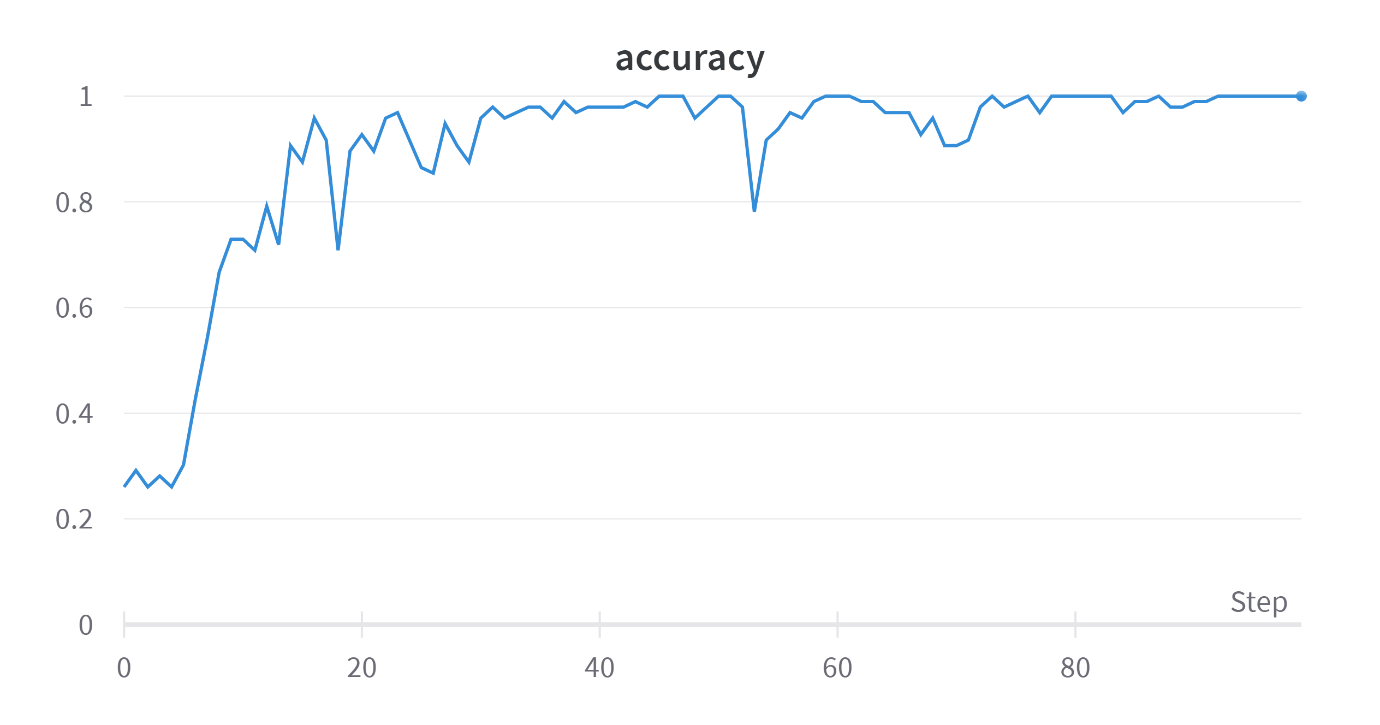


**Steps used for Implementation:**

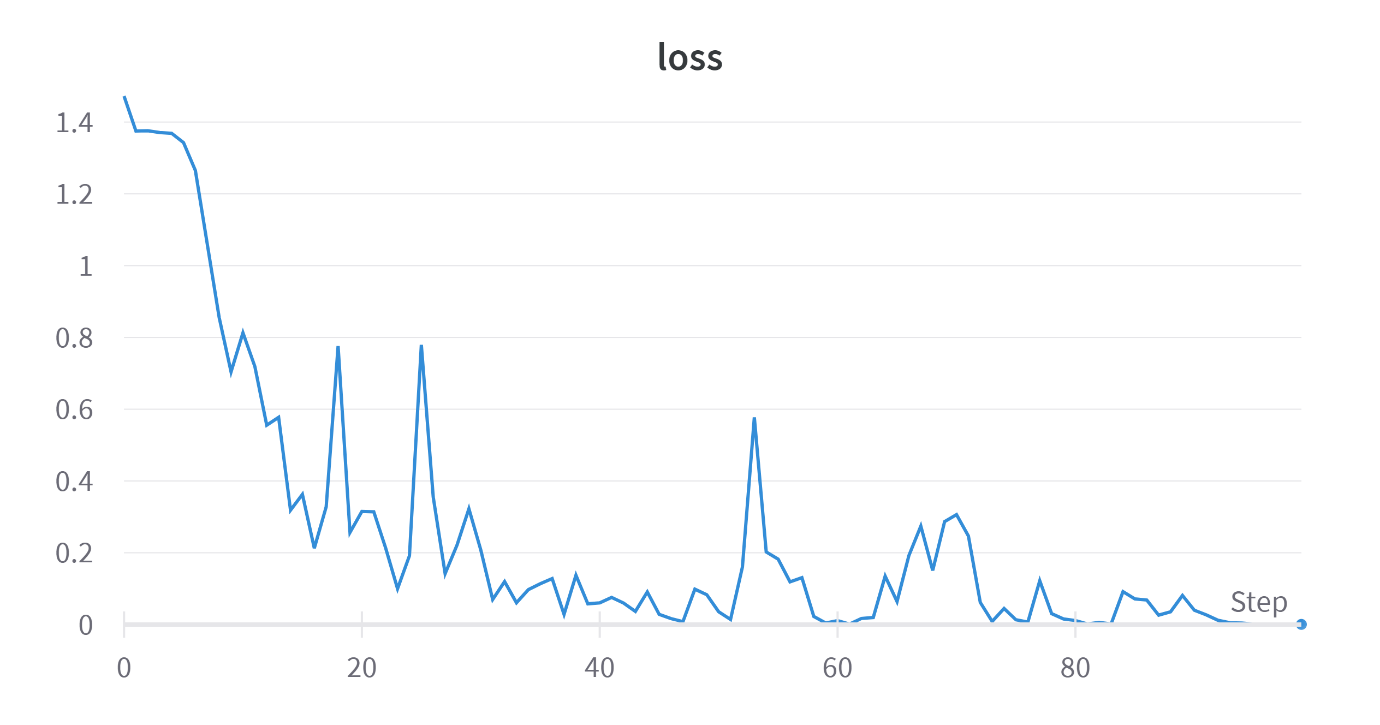
Following are the steps:

1. We first do data processing. We download the required images of the action.
2. Since Dataset contains multiple action classes and we take 20 folders from each action class.
3. We download hd\_kinect\_rgb.mp4, rd\_kinect\_rgb.mp4 videos from each folder in each class.
4. We extract the features and labels of the video and encode it using one hot encoding.
5. After segmentation of videos in its class folders, we need to split the data set into are training dataset, and testing dataset.
6. After that we create a sequential ConvLSTM model, mixed with ConvLSTM layers, max pooling layers, and time distributed layers.
7. We will be using Adam optimizer and categorical cross entropy loss function. Using these we will compile our model.
8. We will use wandb to track the progress of training. So, we will initialize it.
9. Now we will start the training process.
10. After training we will evaluate our model using test data set.

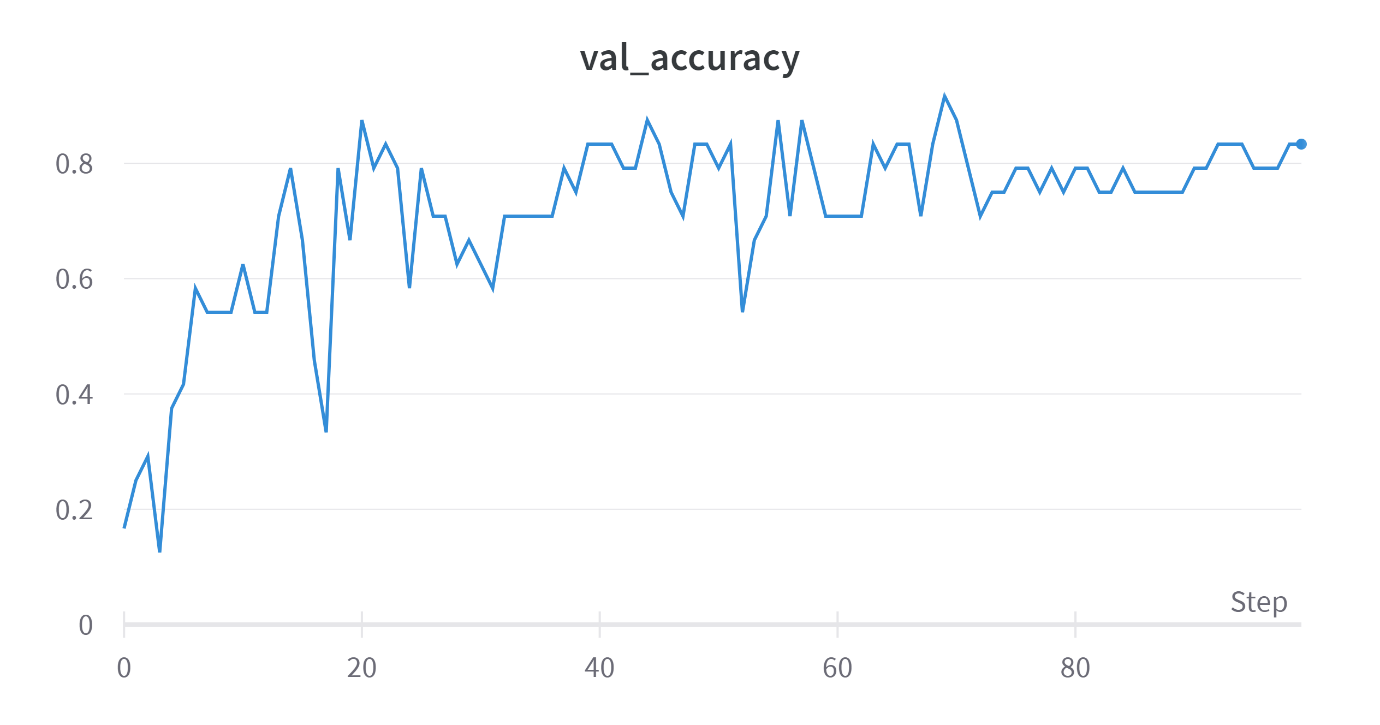
**Accuracy**



**Loss**



**Validation Accuracy**



**Validation Loss**

