Methodology:

* **Dataset collection:**

We will be using the UCF50 – Action Recognition Dataset, consisting of realistic videos taken from YouTube which differentiates this data set from most of the other available action recognition data sets as they are not realistic and are staged by actors.

* **Dataset preprocessing:**

Next, we have performed some preprocessing on the dataset. First, we read the video files from the dataset and resize the frames of the videos to a fixed width and height, to reduce the computations and normalized the data to range [0-1] by dividing the pixel values with 255, which makes convergence faster while training the network.

* **Feature Extraction:**

We have created a function frames\_extraction() that will create a list containing the resized and normalized frames of a video whose path is passed to it as an argument. The function will read the video file frame by frame, although not all frames are added to the list as we will only need an evenly distributed sequence length of frames.

* **Model:**

The two architectures that we will be using to use CNN along with LSTM are:

1. **ConvLSTM :** we have implemented the first approach by using a combination of ConvLSTM cells. A ConvLSTM cell is a variant of an LSTM network that contains convolutions operations in the network. it is an LSTM with convolution embedded in the architecture, which makes it capable of identifying spatial features of the data while keeping into account the temporal relation. For video classification, this approach effectively captures the spatial relation in the individual frames and the temporal relation across the different frames.

we will split our data to create training (75%) and testing(25%) sets. To construct the model, we will use Keras [ConvLSTM2D](https://keras.io/api/layers/recurrent_layers/conv_lstm2d) recurrent layers. The ConvLSTM2D layer also takes in the number of filters and kernel size required for applying the convolutional operations. The output of the layers is flattened in the end and is fed to the Dense layer with SoftMax activation which outputs the probability of each action category.

We will also use MaxPooling3D layers to reduce the dimensions of the frames and avoid unnecessary computations and Dropout layers to prevent [overfitting](https://en.wikipedia.org/wiki/Overfitting) the model on the data. The architecture is a simple one and has a small number of trainable parameters. This is because we are only dealing with a small subset of the dataset which does not require a large-scale model.

Background Study:

CNN:

A Convolutional Neural Network (CNN or ConvNet) is a type of deep neural network that is specifically designed to work with image data and excels when it comes to analyzing the images and making predictions on them. It works with kernels (called filters) that go over the image and generates feature maps (that represent whether a certain feature is present at a location in the image or not) and initially it generates few feature maps and as we go deeper in the network the number of feature maps

Each layer of a ConvNet learns features of increasing complexity which means, for example, the first layer may learn to detect edges and corners, while the last layer may learn to recognize humans in different postures.

LSTM:

An LSTM network is specifically designed to work with a data sequence as it takes into consideration all the previous inputs while generating an output. LSTMs are actually a type of neural network called Recurrent Neural Network, but RNNs are not known to be effective for dealing with the long-term dependencies in the input sequence because of a problem called the Vanishing gradient problem. LSTMs were developed to overcome the vanishing gradient and so an LSTM cell can remember context for

This makes an LSTM more capable of solving problems involving sequential data such as time series prediction, speech recognition, language translation, or music composition. But for now, we will only explore the role of LSTMs in developing better action recognition models. Now let’s move on towards the approach we will implement in this tutorial to build an Action Recognizer. We will use a Convolution Neural Network (CNN) + Long Short-Term Memory (LSTM) Network to perform Action Recognition whileutilizing the Spatial-temporal aspect of the videos.

**Result Analysis:**

In this paper, we discussed several approaches to perform video classification and learned about the importance of the temporal aspect of data to gain higher accuracy in video classification and implemented two CNN + LSTM architectures in TensorFlow to perform Human Action Recognition on videos by utilizing the temporal as well as spatial information of the data.  We evaluated the trained model for ConvLSTM which achieved 80.33% accuracy.