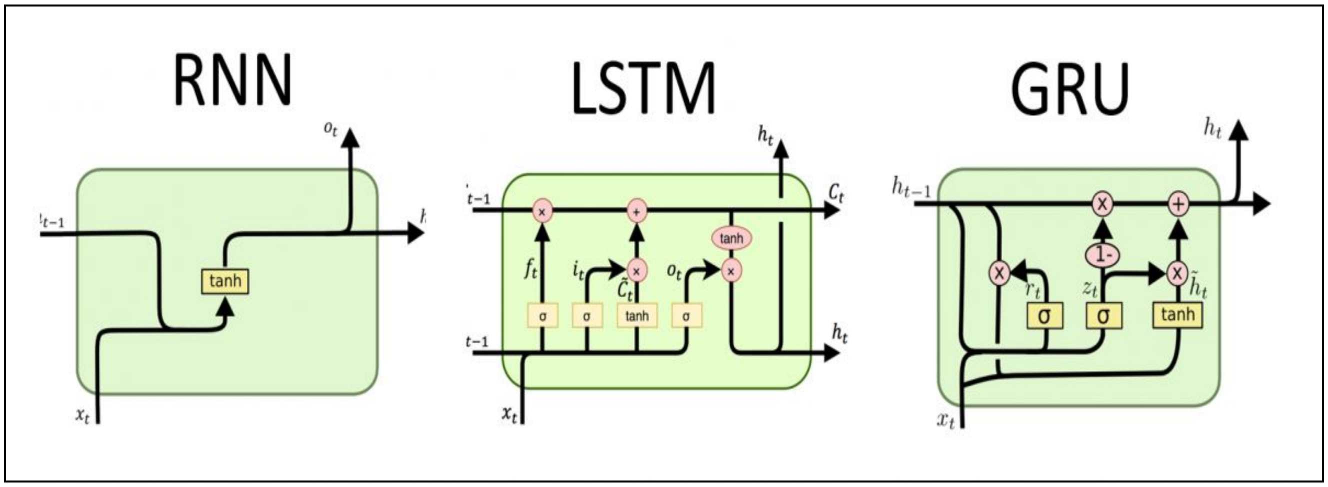
Md_Arif_112669645_hw5.Report

# ***Action Recognition @ UCF101***

## 1. Description

For the task of action recognition, we used the University of Central Florida's Center for Research in Computer Vision's vast UCF-101 action recognition dataset. This dataset consists of short, succinct clips from YouTube depicting actions such as bowling, pullups, applying makeup, and more. It contains about 13,000 videos categorized into 101 action classes. This dataset is commonly used in video related tasks, and thus a lot of work has been published on it. The use of RNN for video classification seems appropriate as videos are a sequence of frames and treating them as a sequence would provide us with more information about the video. Here we don't treat the images as discreet and make use of RNNs and LSTMS. The accumulator value in an RNN is not just influenced by the current input but also the history of the inputs fed to the model. "A Long-short Term Memory network (LSTM) is a type of recurrent neural network that is designed to overcome problems of basic RNNs so the network can learn long-term dependencies. Specifically, it tackles vanishing and exploding gradients."Long short-term memory (LSTM) have achieved great success in processing sequential multimedia data. In this homework we use the Many to One model.

**RNN | LSTM | GRU Cell Structure:**



Tutorial([https://adventuresinmachinelearning.com/recurrent-neural-networks-lstm-tutorial-tensorflow/](https://adventuresinmachinelearning.com/recurrent-neural-networks-lstm-tutorial-tensorflow/)) helps us understand RNN and LSTM Modelling in detail.

## 2.1 Feature Extraction

Preprocessing Challenges and Ways to Overcome:

- Large amount of data makes it very difficult to process the data on all 101 classes. For the first set of modelling we filtered first 25 classes from the data set.
- The creation of features was done in batches. This was done through sorting the dataset and printing the IDs. As Google Colab has limitations on the RAM and GPU Core, I extracted the features on my local system which had higher configuration, it took about 2 days to create the whole feature set. The average speed for this task on a single CPU core was 25* 3 images per minute.
- The video was presampled, preprocessed images in a video and stored them as a tensor. Preprocessing involved cropping and normalising every image and for all the images in a video and creating a 4096-dim VGG features using pretrained VGG classifier's initial layera where we discard the gradients.
- Note: **We did not use the activations of the last layer as the features tend to be task specific towards the end of the network.**

## 2.2 Training Sample Size

A training Sample forms a 3D volume with one dimension encoding temporal correlation between frames and a label indicating what action it is. Thus for every video we have a feature vector of size **25* 4096**.

## 3.1 Model and Performance

We have built our Models using LSTMs and GRU Networks. Given below are the different models run on the data set and their Accuracy observed. Best RNN model achieved a test accuracy of **82.78%** for 25 Classes.

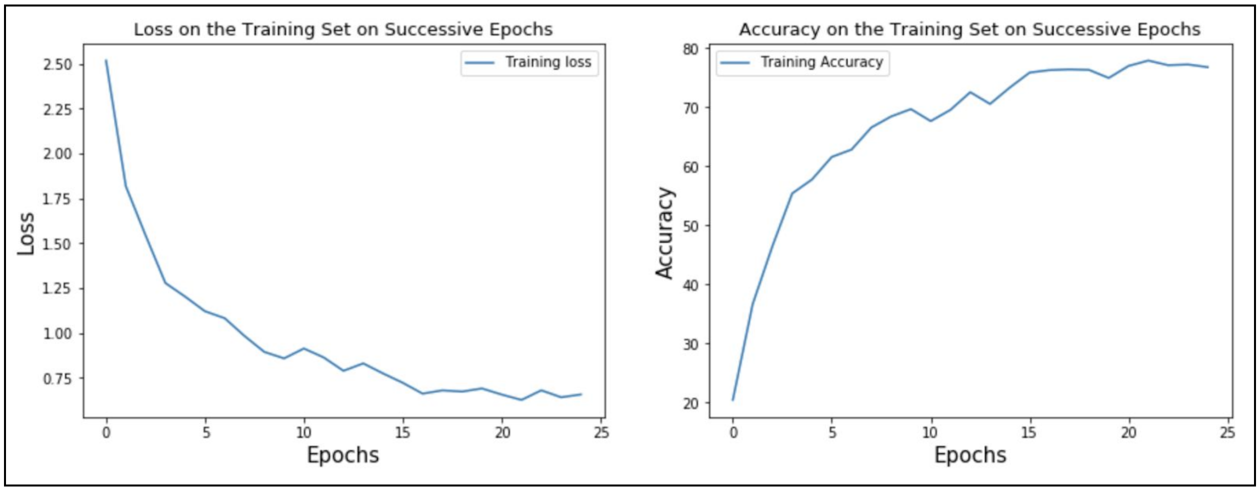


**Fig: Losses and Accuracy observed on the Training Data With RNN GRU**

## 3.2 Model Comparison

| Model Name | Test Accuracy |
|---|---|
| LSTM_RNN @25 Classes | 76.62% |
| GRU_RNN @ 25 Classes | 82.78% |
| SVM @ 25 Classes | 91.34% |
| GRU_RNN @ 101 Classes | 67.13% |
| SVM @ 101 Classes | 69.97% |

- GRU_RNN gives an improved accuracy of ~83% on 25 Classes, whereas 67% on 101 Classes
- SVM on 25 classes observed is 91% whereas 70% on 101 Classes

**4. External References:**

1. https://blog.floydhub.com/a-beginners-guide-on-recurrent-neural-networks-with-pytorch/
2. https://www.semanticscholar.org/paper/Co-Occurrence-Feature-Learning-for-Skeleton-Based-Zhu-Lan/723bd06b14f5fe248955d8d1b3e90b99bf4a9899
3. http://blog.echen.me
4. http://karpathy.github.io/2015/05/21/rnn-effectiveness/
5. http://colah.github.io/posts/2015-08-Understanding-LSTMs/