An Experimental Comparison of Deep LSTM and GRUs for Event Classification in Sports

Kajal Chandani

Snr. 2020244, Anr. 953779

THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN DATA SCIENCE AND SOCIETY

FACULTY OF HUMANITIES AND DIGITAL SCIENCES

TILBURG UNIVERSITY

Thesis Committee:

Prof. Dr. Eric Postma

Dr. Henry Brighton

Dr. J.S. Olier Jauregui

June 10th, 2019

Abstract

Gated recurrent neural networks have shown promising results in sequentially embedded data. They are an improvised version of recurrent neural networks (Hochreiter & Schmidhuber, 1997; Cho et al., 2014). In classification tasks with video data, gated recurrent neural networks are powerful algorithms among all deep learning techniques (Ramanathan et al., 2016). The aim of the present work is to experimentally evaluate the performance of two variants of gated recurrent neural networks, long short-term memory (LSTM) and gated recurrent units (GRUs) in the task of event classification in sports videos. The research question under the present study is:

"How do LSTM and GRUs perform in the task of event classification in sports videos?"

To address the research question, a subset of the basketball video dataset was taken from the study done by Ramanathan et al. (2016). While experimenting with video dataset, the model has to be both spatially and temporally deep because the task involves sequential frames as inputs. For extracting spatial features from 2D frames, VGG16 model was used which is pre-trained on ImageNet weights. The overall architecture can thus be explained as the CNN model (VGG16) for feature extraction and the gated RNNs (GRUs or LSTM) for temporally distributed feature interpretation.

The results of the comparative evaluation between LSTM and GRUs on the basis of accuracy, loss, precision, and runtime per epoch showed that GRUs performed slightly better than LSTM. However, the differences in the results were not very significant, considering the small scale and imbalanced dataset.

Keywords: *Sports event classification, long short-term memory, gated recurrent unit, video dataset*

Table of Contents

1. Introduction

Data-driven technologies have got many emerging applications in sports analytics in recent years. A mere dependency on human involvement in detection and classification tasks has become too old-fashioned. Since deep learning is a very promising data-driven technique, it has been readily applied in the automation of such tasks in today's society.

The extensive broadcasting of sports videos on different transmission mediums needs effective event classification algorithms to reduce manual interpretation of match highlights. To automate event classification on a large video dataset, it is important to consider the most influential algorithm. New results with respect to the application of state-of-the-art methods are announced frequently. An experimental comparison can be used to keep the record of efficiency of these methods in classifying events in sports videos.

The application of deep learning models is widely explored in computer-aided medical diagnoses such as echocardiography (Madani, Arnaout, Mofrad, & Arnaout, 2018), diabetic retinopathy (Petscharnig & Schöffmann, 2018) and gynecological surgery (Mookiah et al., 2013). The major part of processes for developing such computer-aided diagnosis relies on deep learning models. These models are trained in effective classification and prediction tasks using video and image datasets. Looking at the patterns of symptoms from video dataset, automated symptom classification can help in early diagnosis of possible diseases. Apart from the health care field, another social benefit of classifying events using video dataset is human activity recognition (Ghewari, 2017). Human activity recognition can help to prevent fraud and have faster security screening at public places. Looking at the similarities in the above-stated applications, the insights from the present work can be considered in the future to choose between LSTM and GRUs for video-based classifications.

This chapter provides the research objectives in section 1.1 and highlights the research question in section 1.2. Further, section 1.3 and section 1.4 discuss the scientific relevance and main findings of the study respectively. Lastly, section 1.5 briefly shows the approach followed in carrying out the experiment and section 1.6 provides an outline of the thesis.

1.1 Research Objectives

The sequential relationship of frames in videos is the source of rich domain knowledge. It is critical to understand the temporal interactions among them which enable to model and reason the visual relationships (Tsai, Divvala, Morency, Salakhutdinov, & Farhadi, 2019). Apart from understanding the temporal interactions, there are various other time consuming and technically challenging tasks to promote the user-experience of the videos. The list of such tasks includes editing, segmentation, integration, summarization, motion analysis, classification. Specifically, in the domain of sports, event classification has trending applications (Chang, 2019; Ning, 2019; Huang, Li, Zhang, Wu, & Han, 2019; Ramanathan et al., 2016). The retrieval of meaningful events from sports video datasets using both low-level features (for example, motion, trajectories, color) and high-level features (for example, semantic description) has been the topic of research recently (Ramanathan et al., 2016; Nepal, Srinivasan, & Reynolds, 2001; Zhang, Xu, Rui, Wang, & Lu, 2007). Although there has been ample work in semantic description of images through computer vision, sequential relationships in the videos are yet to be explored with the state-of-the-art methods. Due to the complicated spatial-temporal nature of the sports videos, it is challenging for the computer vision to classify the events in a particular match video (Wang, Zhao, & Yuan, 2014).

One of the stepping stones to develop a model that can classify events is the annotation of high-level content (Ramanathan et al., 2016; Saur, Tan, Kulkarni, & Ramadge, 1997). There has been ample development in terms of annotated datasets that can directly be used by researchers to implement video analyzing algorithms. Furthermore, the evolution of various deep learning techniques has resulted in accurate analysis. However, given a visual dataset, a computer like a human, cannot detect an event based on the current frame situation. It has to learn from the sequential pattern of the high-level content in order to classify an event accurately. Therefore, to make the computer learn the semantic reasoning from what has already happened and how does it connect to what is coming up, RNNs are used in various applications (Venugopalan et al., 2014; Donahue et al., 2015). Ramanathan et al. (2016) also used gated RNNs to detect events in basketball match videos. The discussion over this prior research will be done in the later chapters.

In this study, the popular variants of gated RNNs namely GRUs and LSTM are compared through an experiment. Earlier in an evaluation done by Chung, Gulcehre, Cho, and Bengio (2014), the conclusion was made that the gated networks outperform the traditional recurrent units. But while exclusively evaluating LSTM units and GRUs on the tasks of music and speech signal modeling, no concrete conclusion was suggested about which performs better considering their similarities and differences (Chung et al., 2014). The authors also suggested that comparing both the algorithms highly depends on the data selected and the objectives to be achieved. Hence, the objective of the present study is to comparatively evaluate LSTM and GRUs in the task of classifying events in the sports video dataset.

1.2 Research Question

The present study deals with classifying events on the basis of spatial-temporal patterns using GRUs and LSTM. Both of these algorithms being close variants of gated RNNs have been compared in different sequential tasks. However, the challenge still exists in order to evaluate them in the task of event classification in sports videos. Hence, the aim of this study is to empirically evaluate LSTM and GRUs against each other while classifying events in sports videos, such that the most efficient algorithm can be suggested for similar objectives of event classification in future.

Therefore, the research question under study is:

RQ: *How do LSTM and GRUs perform in the task of event classification in sports videos?*

1.3 Scientific Relevance

Despite the complexity of the spatial-temporal nature of sports videos, the idea is that the deep learning algorithm should be able to classify the events in those videos. This ability then solves the issue of traversing the whole video for match highlights and reduces the over-head tasks in sports analytics. For example, a 90-minute long video clip of a basketball match may contain 25-35 events overall, each of them may only take 3-4 seconds approximately. In such scenarios, what we need is an efficient algorithm that could analyze multiple events in a large video dataset and classify them accurately.

Assigning a class label to a particular event in the entire dataset follows the mechanism of supervised learning. However, here the events are embedded in a sequence of time throughout the video of a basketball match, which makes the task much more technically and computationally demanding. Therefore, the nature of the learning techniques to be implemented should be deep hierarchical neural networks. LSTM units and GRUs being the variants of RNNs, have been extensively used in domains of speech recognition, video classification, and video captioning (Khandelwal, Lecouteux, & Besacier, 2016; Staudemeyer & Omlin, 2013; Yue-Hei Ng et al., 2015). Their close and parallel nature have instigated various researchers to evaluate and compare them to see which performs better in various deep learning applications (Chung et al., 2014; Moumen, Chiheb, Faizi, & El Afia, 2018). Since with any given domain of their usage, multiple barriers stand in the way while maximizing the efficiency and power of the model. Therefore, one of the techniques might be superior over the other in terms of its result quality.

To keep updating the scientific academic libraries, it is important to comparatively evaluate the state-of-the-art methods and let the researchers know about which model suits their scenarios. Any potential result based on the experiment in this thesis can be a contribution to the application of deep learning techniques in classifying events in sports videos and later can be applied on videos from different domains as well.

1.4 Approach

The main approach of the present study is to compare LSTM and GRUs. The experiment is done on the basketball video dataset, each of the match videos being approximately 1 hour and 30 minutes long. There are around 40 events annotated in every match. The data is prepared in such a way that only the video clips of events are extracted out of the entire match. These video clips are then divided into training, validation and test dataset.

Further, the sequential frames are extracted considering that the sequential nature of the data is essential to solve the task of event classification. For a model to learn from the spatial and temporal dynamics of these frames, dimensional features are extracted which will be further explained in chapter 2 and 3. The deep learning models of LSTM and GRUs are fed with the extracted features. This way the temporal features are combined with the spatial features in 2D image representations in sequence. Then the variants of RNNs are compared against each other based on their performance while classifying an event.

1.5 Thesis Outline

Chapter 2 of this thesis provides the background of the techniques used in the experiment. It elaborates on the related studies on deep learning and describes the variants of the RNNs used and compared in this thesis. It also describes the earlier studies where researchers worked on classifying events in the sports videos with other deep learning techniques. Chapter 3 will provide more details on the dataset and how it was processed before using in the model. Chapter 4 describes the methods and the experimental setup. It also explains the evaluation metrics considered to compare the models used in the thesis. The results of the performed experiments using both models will be given in chapter 5. Chapter 6 will discuss the points of improvement in future research. Lastly, chapter 7 will conclude the thesis with an answer to the research question.

1.6 Main Findings

The main finding of this thesis is the full-fledged set-up that starts at video dataset preprocessing and feature extraction using VGG16 to prepare the input for the event classification model. Then this input is used for LSTM or GRUs, considering spatial-temporal features of video dataset. The results in terms of accuracy through LSTM and GRUs are 10% and 14% above the baseline, respectively. Further, the study also comparatively evaluates the LSTM and GRUs on the basis of loss, precision, and runtime per epoch. These scores obtained by adding GRU layers are slightly better as compared to the model with LSTM layers.

2. Related Work

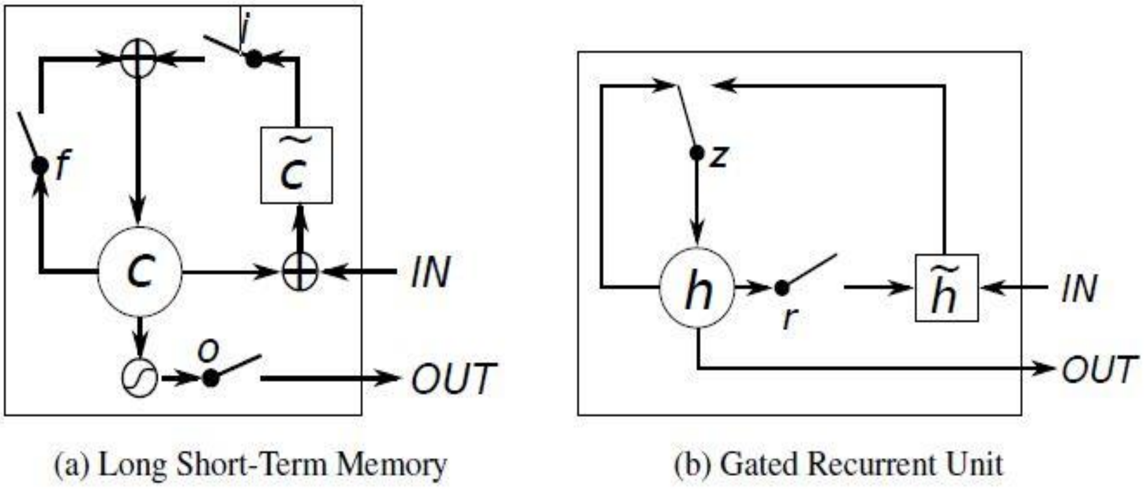
This chapter provides a background to the previous researches related to the current study. The first section provides an overview of how Gated Recurrent Neural Network has given promising results in several applications in sports and motivation they provided for this study. The subsections bifurcate the analysis of the performance of LSTM units and GRUs in various researches done. In the second section, the studies done so far on the task of event classification in sports are discussed. Lastly, the final section discusses the researches where GRUs and LSTM were comparatively evaluated in terms of different applications.

2.1 Gated Recurrent Neural Network

In recent years, there have been many research efforts done in the domain of sports. The results not only aim at better handling of sports data but also on enhancing the viewers' experiences. The advancements in the field of sports analytics using deep learning have shown plausible results in players' training, ceasing possible injuries, and shot predictions. With an edge of numerous interesting results provided by deep learning researchers, visual learning has been able to gather more attention. Visual learning on inputs with only spatial information is much explored using deep convolutional neural networks and they are powerful enough in the field of computer vision (Géron, 2017). One of the constraints that CNNs are not used for video-based inputs is that they allow the input of a fixed-size vector with only spatial information (Karpathy, 2015; Géron, 2017). Although 3D-CNNs accept video inputs, the implementation is computationally demanding (Karpathy et al., 2014)

Hence, here comes the role of recurrent neural networks. The RNNs allows processing with sequences of input vectors thus capturing the temporal features as well. Karpathy (2015) in his blog on RNN explained the relation of a sequence of inputs with the output. Karpathy (2015) said, "Output vector's contents are influenced not only by the input you just fed in but also on the entire history of inputs you've fed in in the past". Therefore, RNNs are capable of handling sequential data while also storing temporal information about the prior states. This enables us to imagine a network which is not organized in a straightforward manner because it's not one-way anymore (Karlsson, 2017).

Specifically, due to the ability of RNNs in storing historical or rather temporal information, it is highly rated in tasks of speech processing and recognition (Graves, Mohamed, & Hinton, 2013), handwriting recognition (Pham, Bluche, Kermorvant, & Louradour, 2014) and scene labelling (Liang, Hu, & Zhang, 2015). Regardless of RNNs being capable of learning from the past, they cannot perform well while learning from multiple time-steps due to the vanishing gradient problem (Géron, 2017). Due to this problem, the networks take much longer training time and eventually lose a lot of information. This is because, with every recurrence, the memory of the first input fades away. Hochreiter and Schmidhuber (1997) came up with the solution to this problem by introducing the first gated recurrent neural network called long short-term memory, ever since used for various sequential datasets. Another variant of gated RNNs called gated recurrent unit which worked on similar base principals was introduced by Cho et al. (2014).



(a) Long Short-Term Memory

(b) Gated Recurrent Unit

Figure 1: Illustration of (a) LSTM - i, f and o are the input, forget and output gates, respectively. c and ~c are the memory cell and the new memory cell content (b) GRUs - r and z are the reset and update gates, and h and ~h are the activation and the candidate activation. (Figure illustrations are taken from the paper written by Chung, Gulcehre, Cho, and Bengio, 2014)

The LSTM architecture has got three gates: input, forget and output gate. Géron (2017) explained the working of these gates as follows. The input gate decides if the new input should be allowed in the network. The forget gate determines if the information should be stored or not based on its importance. Finally, the output gate receives the impact at the current time-step. On the other hand, GRUs have update and reset gates. The former controls “how much the unit updates its activation, or content” and the latter allows the unit to start fresh as if it's receiving the first input from the sequence and forgetting the previous ones (Chung et al., 2014). GRUs, unlike LSTM, do not have separate memory cells.

To sum up the architectural differences in LSTM and GRUs, it can be said that LSTM provides controllable exposure to memory content, whereas GRUs do not control and expose the entire content (Chung et al., 2014). Both of these variants of gated RNNs are popularly applied to video-based dataset on several tasks including sports analytics. Focusing on sports analytics and related work on video classification, the applications of LSTM and GRUs studied in recent literature are discussed next.

2.1.1 Long Short – Term Memory

The anticipation of the end results of a sequence task is a simple phenomenon for the human mind. Humans learn from experience and make predictions based on what they know or remember from their past with added logical reasoning. Researchers understand that if a machine has to learn to predict the future, it should know and remember what can generally lead to a possible outcome. For example, if the neural network has to predict the possible accident with an automated car, it should know where the car was in the last few frames (Géron, 2017). Through LSTM units, the hidden recurrences facilitate the decision of what to remember and the input recurrences focus on remembering the immediate previous entry. What it stores in the memory is the combination of the input data and the knowledge from the previously hidden layer. The input layer then generates the hidden layer using the method of forward propagation (Ramanathan et al., 2016).

In the domain of sports analytics, LSTM has been used as both stand-alone and also in multiple combinations of neural network layers to perform various tasks like object tracking, sports video summarization, video captioning, attention models and so on. In the task of action classification in sports, Ullah, Ahmad, Muhammad, Sajjad, and Baik (2018) also coupled the feature extraction layer preceding the training of LSTM model. They used pre-trained AlexNet (Krizhevsky, Sutskever, & Hinton, 2012) for feature extraction and architecture of deep bidirectional LSTM (Graves, Fernández, & Schmidhuber, 2005). The results of the study done by Ullah et al. (2018) showed 91.21% accuracy in UCF101 dataset and 87.64 % in HMDB51 dataset, claiming that the results outperformed CNN based methods. Another demonstration of CNNs and LSTM, titled as LRCN was done by (Donahue et al., 2015), showed good results in the domain on activity recognition. LRCN was later adopted by Karlsson (2017) for the purpose of video classification in sports, achieving 89% accuracy in classifying sports category.

Ramanathan et al. (2015) worked on identifying the key-player in each frame of an event in basketball dataset. Key-player identification was an adjacent consequence of bidirectional LSTM to classify an event at the frame-level. In the current study, rather than capturing the features in a single frame level, the features for the entire sequence are captured for a particular event. These features are then given as an input to the LSTM layer, (discussed more in the *Experiment* chapter). Ramanathan et al. (2016) also suggested that the temporal flow of the frames form a solid base for the model to learn about the connection between the spatial features. Since LSTM units are popular for utilization of the memory cells storing the sequential information, the evaluation in terms of providing a sequence of features might be insightful for further scientific explorations.

2.1.2 Gated Recurrent Units

Gated recurrent units are another variants of gated RNNs proposed by (Khandelwal, Lecouteux, & Besacier, 2016), which are designed to be simpler than LSTM in terms of computation. While discussing in terms of simplicity, GRUs do not incorporate memory cells, they have reset gates instead. These reset gates allow the hidden state to leave the unimportant information and thus, focusing on the *quality* of content. Further, the update gate controls the flow of information from the hidden state in terms of *quantity*, which makes the process compact (Cho et al., 2014). Though GRUs act the same as LSTM in terms of learning the long term dependencies, they are expected to perform efficiently in terms of computation (Khandelwal et al., 2016).

GRUs have been so far used in many prediction and classification tasks where they were able to outperform many traditional models. For example, auto regressive integrated moving average (ARIMA) model was outperformed by GRUs for traffic flow prediction (Fu, Zhang, & Li, 2016). Other examples where GRUs performed better than state-of-the-art RNNs and LSTM models are session recommendation system (Hidasi, Karatzoglou, Baltrunas, & Tikk, 2015); language modeling (Moumen et al., 2018) and sequence modeling (Chung et al., 2014).

In the task of video classification, Zhu, Xu, and Yang (2017) used multi-rated GRUs as encoders to encode the frames of YouTube2Text video dataset. Their model coupled with Res-Net-200 features produced an accuracy of 34.45% on Meteor which was comparatively higher than CNNs and LSTM. The data they experimented had variable input length of sequences similar to the dataset used in the present study.

There are multiple scenarios where LSTM has been applied over sports dataset in order to retrieve results on action recognition of players, event detection and classification. But as per my knowledge, GRUs have not been explored in classifying sports videos on the basis of events or shots played in a match. The architecture sophistication of VGG16 – GRUs for video classification is also yet unknown. Therefore, the present study works on exploring the performance of GRUs as compared to that of LSTM while receiving inputs from the pre-trained VGG16 model for sports event classification.

2.2 Background on Event Classification in sports using Deep learning techniques

An event in sports is a semantically significant moment captured in a particular portion of the entire video. Understanding the temporal and spatial features in a sports event can be inspired by multiple studies done in the field of action recognition (Ramanathan et al., 2016). Various studies have been done on recognizing the actions in videos and classifying the type of sport using the dataset released by Rodriguez et al. (2008). Karpathy et al. (2014) used this dataset for sports classification where they applied combinations of fusion techniques using CNNs with one frame at a time. On the other hand, for a similar task, Fernando and Gould (2016) used rank pooling method. Both of these methods achieved an accuracy of approximately 80-85%. According to Ullah et al. (2018), the RNN architectures overshadow the utilization of CNNs for a similar task since RNNs combine the hidden patterns in the temporal-spatial data. Furthermore, it was found that stacking the pre-trained CNNs to extract features and RNNs for temporal handling is a state-of-the-art approach for classifying videos (Karpathy et al., 2014; Donahue et al., 2015; Ramanathan et al., 2016; Ghewari, 2017; Karlsson, 2017).

One of the most recent works done by Yu, Lei, and Hu (2019) demonstrates a model that was able to detect events in the soccer video match through play and replay clips, making it a valuable contribution in the field of video classification. For replay detection, the researchers used VGG16 which was trained to recognize the logo that appears between a play and replay clip. Yu et al. (2019) also implemented LSTM followed by multi-feature fusion to extract spatial-temporal features and therefore, achieving good performance in the replay detection. The dataset they used was based on multiple temporal scales similar to the dataset used by Ramanthan et al. (2016). The data with multiple temporal scales has no negative impact on the performance of gated RNNs (Karpathy, 2015; Géron, 2017; Ramanathan et al., 2016; Yu et al., 2019). Because of that, the present study uses the subset of basketball video dataset created by Ramanthan et al. (2016) to make the comparative evaluation in LSTM unit and GRUs.

2.3 Comparative Evaluation of LSTM units and GRUs

As discussed in section 2.1, while working with a video data set, it is essential for the recurrent networks to understand the sequential nature of the inputs. Other than video datasets, researchers have also studied the relevance of gated RNNs in the sequential dataset such as continuous vocabulary inputs (Khandelwal et al., 2016), polyphonic music and speech signals (Chung et al., 2014).

In the research done to evaluate the performance of GRUs and LSTM in the domain of large vocabulary continuous speech recognition, it was found that GRUs performed better than LSTM in terms of computational efficiency since, GRU is a simpler model (Khandelwal et al., 2016). The authors used the dataset from TED talks, which is more complex than just reading the text transcripts. In their experiment, Khandelwal et al. (2016) found that given the same number of parameters in both GRUs and LSTM, GRUs perform better than LSTM units. In addition, they also saw that deeper GRUs outperform simple GRUs and LSTM units. Khandelwal et al. (2016) also suggested that the combination of CNNs and gated recurrent layers could form an appropriate architecture for classification tasks. Essentially for the task video classification, training on motion knowledge and spatial dimensions has been a popular approach which is also implemented in the present experimental evaluation. Chung et al.

(2014) suggested that the nature of dataset has also a major impact on the performance of GRUs and LSTM.

Furthermore, Miech, Laptev, and Sivic (2017) compared LSTM and GRUs for video classification using the gated units for temporal aggregations on the assumed features on Youtube-8M large-scale videos. The authors found that taking sequences of frames in temporal or random order brought no differences in the performance of the LSTM units and GRUs. However, in the present study, the spatial information is only processed in the temporally ordered sequence.

GRUs and LSTM units were also compared in Arabic language diacritization in terms of their training runtime and the error rate (Moumen et al., 2018). Though the accuracy given by GRUs was comparable to LSTM units, the runtime efficiency was increased by 18.2% while using GRUs. The process and evaluation metrics used to compare LSTM and GRUs in event classification task in the present study is illustrated in chapter 4.

3. Dataset Description

The dataset consists of NCAA basketball match videos. These videos have dense temporal annotations with respect to 11 classes of events taking place in a match. The pre-classified 11 events are 3-pointer success and failure, free-throw success and failure, layup success and failure, other 2-pointer success and failure, slam dunk success and failure, and steal success. The data used in the present study was taken from the study done by Ramanathan et al. (2016), used for detecting events and key players in the basketball videos using deep learning techniques. The authors collected approximately 1.5 hours of lengthy videos from YouTube and manually labeled the identified events. This data can be established as the ground-truth in the present study. The data preparation and preprocessing steps that were performed in this study are described in the subsections below.

3.1 Data preparation

Due to computational boundaries, the experiment was done on a portion of data introduced by Ramanathan et al. (2016). The total number of video clips used in the experiment was 501. The selection of the dataset was randomly made due to which one of the 11 classes (steal success) got missed as it was frequent. Henceforth, everywhere in the study, the experiment was implemented on the remaining 10 classes.

For the task of data preparation, the first step was to download the videos from YouTube. The spreadsheet provided by the researchers Ramanathan et al. (2016), has the YouTube IDs used for downloading the videos. The task of downloading the videos was done using pytube v9.5.0 library.

Additionally, the spreadsheet also contains the start and end timestamps (in milliseconds) in which each of the events took place throughout the match videos. These timestamps were used to slice the match videos and store them in the form of video clips pertaining to an event. As defined by Ramanathan et al. (2016), an event is a moment when the ball leaves a player’s hand and lands somewhere (in/out of the basket). The slicing was done using FFMPEG tools of moviepy python modules which is famous for handling videos. The pre-defined function called ffmpeg_extract_subclip used timestamps (end-time and start-time) as parameters along with the video file. The video-clips for each interval were thus stored in the directory corresponding to the parent video file.

The next step was to convert the video-clips into a sequence of frames. Here, OpenCv2 library is used to convert the videos into frames. The frame rate per second was 25, so the maximum number of frames per event was 100 because each event took a maximum of 4 seconds (Ramanathan et al., 2016). Furthermore, to simplify the data preprocessing steps, the sub-clips extracted as events were stored in the newly created directories using miscellaneous operating system interfaces library of python. The storing of video-clips were planned in a manner that each of them got stored in the folders named as per their class.

Finally, the last step for data preparation was to divide the data into train, validation and test set. The split resulted in a total of 277 training, 116 validation and 75 test video-clips, each of which has one of 10 class labels. The code written to process the split also stores the 10 class folders in each of the train, validation and test folders as parent folders.

3.2 Data preprocessing and Spatial-temporal Feature Extraction

The preprocessing of data included the extraction of spatial and temporal features of the sequence of frames. In the study done by Ramanathan et al. (2016), Inception 7 networks were used to extract the features from the 2D images. Since CNNs are the state-of-the-art method for feature extraction (Simonyan & Zisserman, 2014), in the present study, VGG16 has been used for the same. According to Acharya, Yan, and Khoshelham (2018), VGG16 architecture of CNNs was the first one to beat human recognition in ImageNet dataset.

The VGG16 network is pre-trained on large-scale image classification task as the part of ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al., 2014). The network presented a top-5 accuracy of 92.3% on ImageNet dataset which has 14 million images classified among 1000 classes. Thus, VGG16 is able to produce a rich spatial feature representation. The architecture of VGG16 is a 16 layers deep network, including 3×3 convolutional layers, stacked one over the other in increasing depth. Then max-pooling layers are added to reduce the volume size. Finally, there are two fully connected layers and the softmax classifier. The width starts at a size of 64 for every network and increases by a factor of 2 after every sub-sampling or pooling layer (Wang et al., 2017).

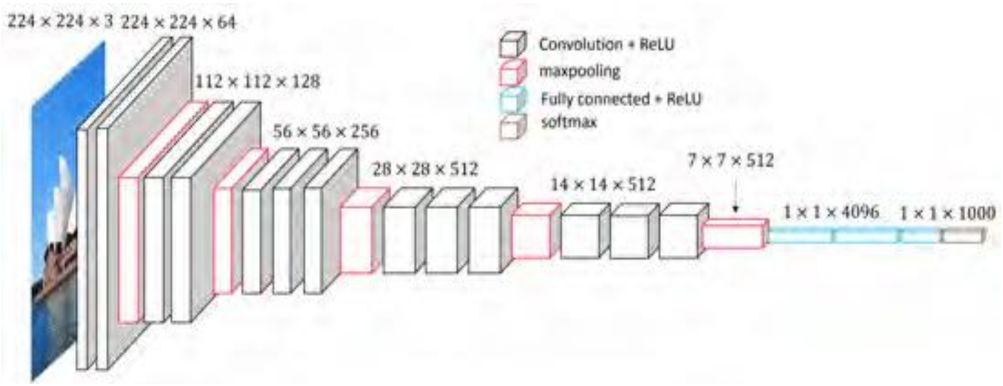


Figure 2: VGG16 architecture, a figure is taken from a paper by Acharya, Yan, and Khoshelham, (2018) explained on the basis of the description given by Wang et al. (2017).

For inputting the frames to the VGG16 network, the prerequisite is to resize the input images to spatial dimensions of 224 x 224 pixels. After resizing, the pixels are converted to a NumPy array for each image. Further, a function was defined to convert the input into 224 pixels wide, 224 pixels tall, having 3 channels (one for each of the Red, Green, and Blue channels) as an array of 224 x 224 x 3. In order to apply VGG16 network, the dimensions obtained from the previous step was expanded to the dimension of 1 x 3 x 224 x 224. Finally, the features were extracted using the pre-trained ImageNet weights through VGG16 network.

The output of this last step was an array of dimensions 7 x 7 x 512. This output was then converted to a vector by flattening it. The next and most crucial task for the data preprocessing was to generate the features of all sequential frames and store them efficiently to be used to classify events. For this task, the functions were defined which created a similar structure of folders as explained in the part of data preparation. The feature vectors of the sequence of frames of a particular video-clip were stored in a text file, in the folder corresponding to its class label. To map the labels, the technique of One Hot Encoding was implemented. Thus, the features from a sequence of frames were processed to be used as the input for LSTM and GRUs.

4. Experimental Setup

This chapter describes the experimental methodology carried out in this thesis. First, the software, hardware and the algorithms used are discussed. Subsection 4.2 describes the input required and output expected to fulfill the objective of event classification in sports videos. The task of event classification is discussed while taking the support of available researches. Subsection 4.3 introduces the baseline formed on minimal probabilities. Subsection 4.4, elaborates on how the models LSTM and GRUs are trained using the video frames. The chapter is concluded in subsection 4.5, which elaborates the evaluation metrics used to compare the two models in the task of event classification.

4.1 Software, Hardware, and Algorithms

This study is set up to evaluate the empirical performance of two deep learning algorithms GRUs and LSTM in terms of their capability of classifying events in the basketball videos. Thus, the algorithms mentioned are in the centric focus.

Python was principally used as the language of development. The development was done on the collaborator free cloud services provided by Google research, which also supports free Jupyter environment and faster GPU. The process required installation of Keras for deep learning; Tensorflow in order to parallelize the computation of the GPU. Further, various libraries were used at different stages of the process. These libraries were Pytube for downloading YouTube videos, MoviePy for converting videos into sub-video clips as per the timestamps coded, Opencv2 library for converting videos into frames, and Matplotlib for required data visualization. The other libraries used for supporting programming tasks were numpy and pickle.

4.2 Explanation of the task of event classification in sports videos

The video dataset created by Ramanathan et al. (2016) has a variable length of events. The maximum length of an event happened in a match is kept 4 seconds. Since the frame rate per second is 25, the maximum length possible is 100 frames. The spatial features extracted using VGG-16 network were aggregated into sequences of features for each event corresponding to one of the 10 classes. Hence, the input for the models was the aggregated sequence of features in the form of a vector. Given that the vector length varies due to variable length events, the input sequences were padded equivalent to the maximum length so that they remain consistent for the model. Eventually, the 10 output classes were one-hot encoded. The task thus can be explained as one-of-many classification where each event can belong to one of the classes. The number of output neurons is equivalent to the number of event classes and can be gathered in a vector, whereas, the ground truth vector represents a one-hot encoded vector of events.

4.3 Baseline

In the present study, there are 10 possible classes, namely 3-pointer success, 3-pointer failure, free-throw success, free-throw failure, layup success, layup failure, other 2-pointer success, other 2-pointer failure, slam dunk success, slam dunk failure.

Table 1. Training sample distribution across event classes

Event Classes	**Training sample distribution**
3-pointer success	37
3-pointer failure	22
free-throw success	18
free-throw failure	7
layup success	48
layup failure	34
other 2-pointer success	53
other 2-pointer failure	52
slam dunk success	3
slam dunk failure	3

Since the data was randomly portioned, and also due to the nature of the original dataset, the classes are highly imbalanced. For framing the minimal accuracy to form a baseline, the class with a majority number of instances is used as a reference. Other 2-pointer success being the majority class has 53 out of 277 occurrences. Therefore, if we consider that the baseline classifier always predicts the event to be other 2-pointer success, then the accuracy of the baseline would be 19.13%. This minimal accuracy will be used as a reference to check the performance of the trained models in the task of event classification.

4.4 Training the Models

After the input features are padded to the maximum length, the input for the GRUs and LSTM layers has to be reshaped because the layers accept the input in a three-dimensional shape. The dimensions in the 3-D input represent the samples, time-steps and the features. Here, a sample represents one sequence, a time-step is one point of observation and the features represent one observation at a given time-step. Therefore, the input layer of LSTM and GRUs assumes that the training dataset has 1 or more samples, has a specification about the number of time-steps and the number of features. According to Ullah et al. (2018), "training large data with complex sequence patterns (such as video data) are not identified by the single LSTM cell". In this experiment, the gated RNN units are stacked (in a form of multi-layered RNN) in order to learn the long term dependencies. The corresponding next layers receive information of hidden states as input. The input layer in the current setting receives an array of shape 10 * 100 * 25088 where 10 is the batch size, 100 is the number of time-steps (as the result of padding) and 25088 is the length of the sequence which is generated from the output of VGG16 features.

Any deep neural networks such as LSTM and GRUs are more likely to have a problem of overfitting (Wang, Chu, Liu, & Zhou, 2017). Therefore, it is possible that the model results in much lower accuracy on the holdout dataset as compared to the over-fitted training data. Thus, the *dropout* layer provides regularization and decreases the probability of overfitting by dropping units that can adapt more than required (Wang et al., 2017). The dropout layer is stacked before the first hidden layer with 20 hidden units and preceded by the two LSTM or GRU hidden layers with 10 hidden units.

The next layer is the dense layer with *ReLU* or rectified linear unit activation function because it is fast and controls the overfitting of data (Mao et al., 2014). Another dense layer with a softmax function is used. The *softmax* function is common in neural network based classification processes. It is used in the final layer, giving the output that represents the probability distribution over different possible classes. The model has been fitted with a callback to the time history for the time plot. Also, the class weights are explicitly defined in order to tell the model to pay attention to the class samples that have extremely low occurrences.

To compile the model, *categorical cross entropy* has been used to measure the error rate or the loss incurred in the distribution of prediction with respect to actual classes. Furthermore, to measure the accuracy, *categorical accuracy* has been used because the problem is dealing with multiple classes. The effective *adam* optimizer is used which is known to deal with sparse gradients and non-stationary objectives (Kingma & Ba, 2014).

There was a need for trying out values for a few important parameters that were essential for final training before the evaluation. This process included finalizing among the choices of adaptive learning rates, values for the number of training epochs and the mini-batch sizes. Ultimately, the value for batch size was fixed to 10 and the training was done for 50 epochs. The figures for the models generated using TensorBoard are displayed below.

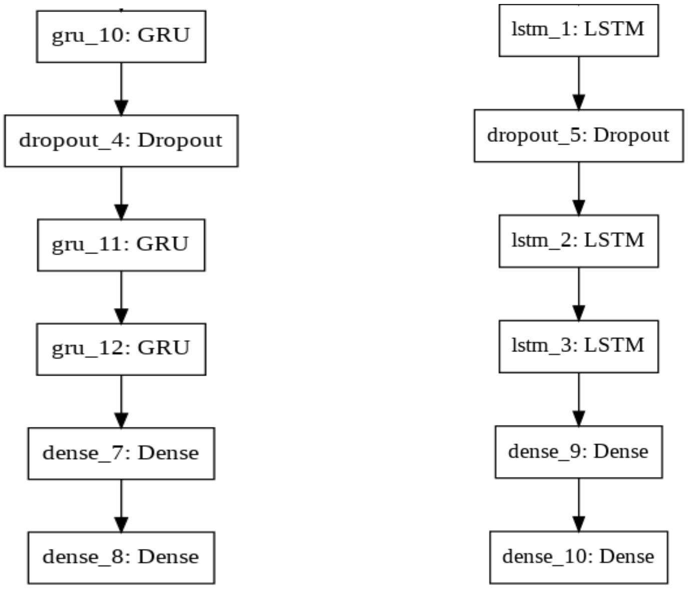


Figure 3: Illustration of (a) GRUs (b) LSTM model architecture used in training for event classification as per the network explained above

4.5 Evaluation

The following metrics will be used to evaluate LSTM and GRU models.

4.5.1 Accuracy

For the evaluation of the models, the mean class accuracy (MCA) is selected. MCA is calculated by weighing the accuracy of individual class as per the ratio of the event instances present in each class. This is done by assigning class weights while fitting the model and the metric used is "categorical accuracy". This method is chosen to evaluate the models because of the inconsistency in the distribution of a particular type of event, which can otherwise have a disproportionate impact on the accuracy rate.

4.5.2 Categorical cross-entropy loss

Multiclass classification is classifying training samples into multiple categories (one class-of-many samples). Since we have been using the softmax activation, categorical cross-entropy loss is more suitable than any other entropy loss function. Monitoring the loss along with the accuracy can indicate the fitting of the validation dataset.

4.5.3 Precision

Precision is another interesting metric to look at while doing a classification task, especially when it is a matter of imbalanced classes. It has also been reported in the event classification task done on the same dataset by Ramanathan et al. (2016). To find out how much of the predicted classes are correct when compared to the class labels present in the ground truth, the precision score gives better insights on the performance of the models.

4.5.4 Training Runtime

Lastly, the training runtime of both models is recorded in terms of a number of seconds per epoch. This is to give the intuition about which model is computationally efficient with respect to the amount of data considered in this particular study.

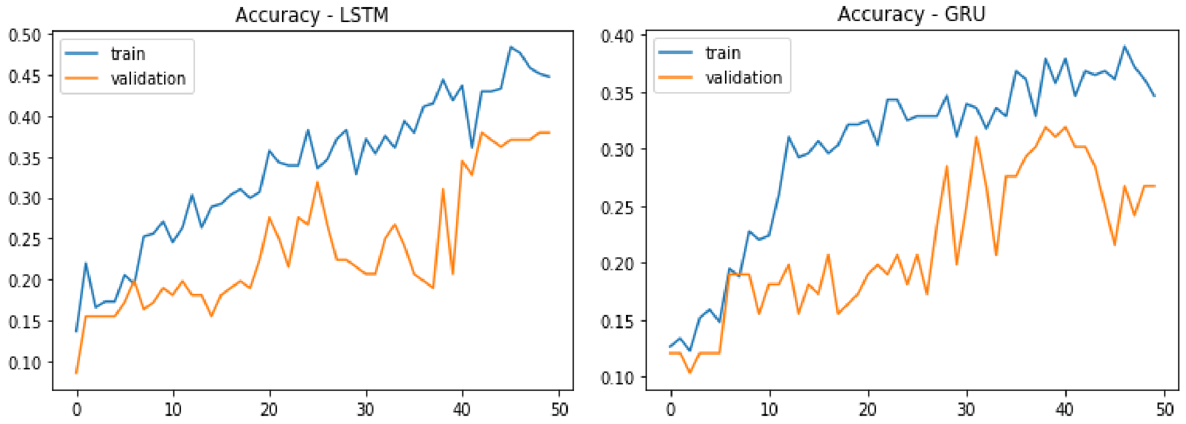
5. Results

This section will report the results of the experiments based on the task of event classification. The tables and figures as per the evaluation metrics will be illustrated for both of the models. The comparison of the models with the baseline is made in terms of mean class accuracy. To evaluate which of the training models performed better overall, loss, precision, and training runtime are examined.

With different dropout rates, Table 2 shows the classification performance on mean accuracy and Table 3 shows the weighted entropy loss for both the models. Based on the best dropout value, accuracy and weighted entropy loss are plotted against per epoch in figure 4 and 5, respectively.

Table 2. Overview of Accuracy for baseline, LSTM, and GRU with varied dropout parameters

	Baseline	LSTM			GRU		
		Dropout rates			Dropout rates		
		0.3	0.5	0.9	0.3	0.5	0.9
Accuracy	0.1913	0.293	0.284	0.293	0.336	0.207	0.121



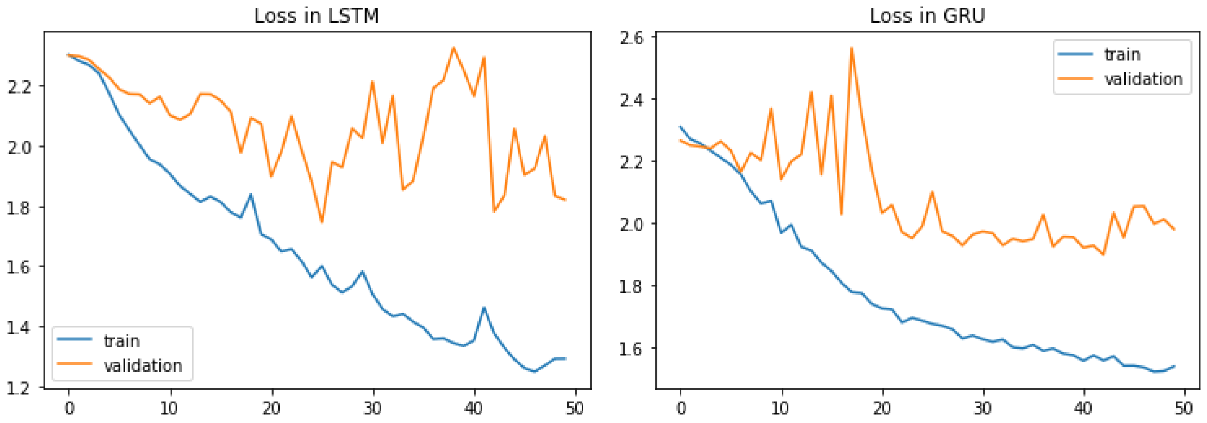
(a) *(b)*

Figure 4: Illustration of Accuracy per epoch (a) LSTM (b) GRUs model (dropout rate – 0.3)

As shown in figure 4, the training accuracy and validation accuracy are randomly fluctuating with increasing behavior overall. In both the models, training accuracy turns out to be larger than the validation accuracy at each epoch, suggesting the sign of overfitting. This might be due to the small data size. However, it can be seen that even with 50 epochs, the training accuracy has an increasing trend. Therefore, to improve the accuracy further, a number of epochs can be increased. In terms of validation accuracy, there is a sudden decrease in GRUs model during the last epochs.

Table 3. Overview of cross-entropy loss for LSTM and GRUs with varied dropout parameters

	LSTM			GRU		
	Dropout rates			Dropout rates		
	0.3	0.5	0.9	0.3	0.5	0.9
Weighted entropy loss	1.911	2.027	2.232	1.897	2.584	2.247



(a)

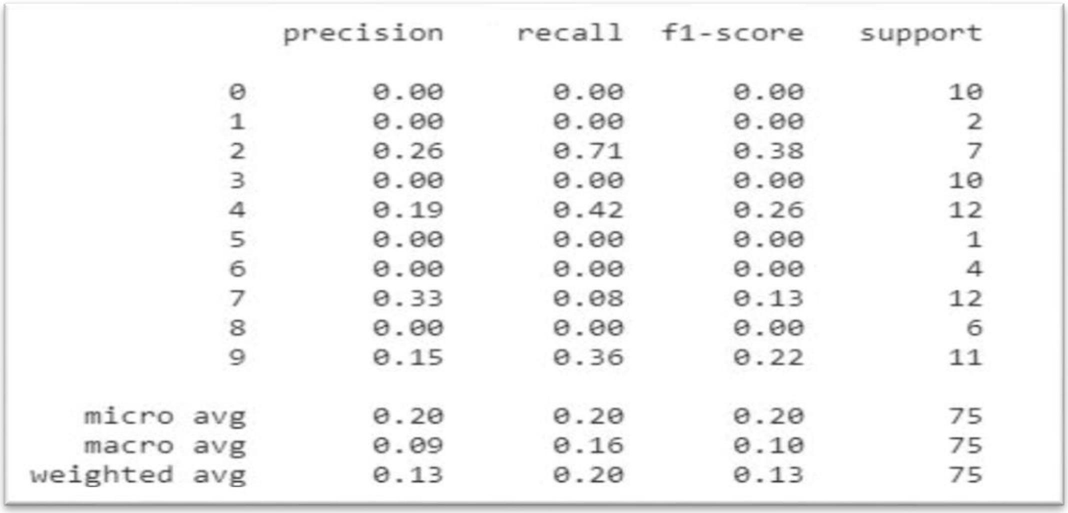
(b)

Figure 5: Illustration of loss per epoch (a) LSTM (b) GRU model (dropout rate – 0.3)

From figure 5 above, it can be seen that the weighted entropy loss while training is much less than the weighted entropy loss on validation data, hence, indicating overfitting of the models.

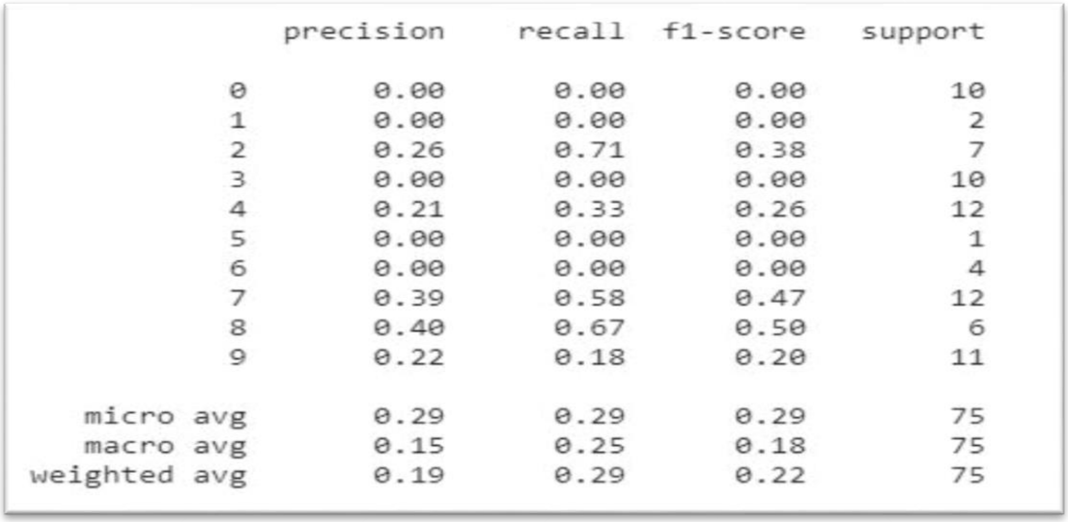
Among the different dropout rates for both the models, a dropout of 0.3 represents the best performing models. Though LSTM had the same accuracy while the dropout rate was 0.3 and 0.9, the weighted cross-entropy loss considerably increased by 1.312. As compared to the baseline, the most suitable model came out to be GRUs (dropout rate 0.3) with an accuracy of 33.6% and incurred a loss of 1.897 (shown in Table 2 and 3 above). Henceforth, the evaluation metrics will be discussed based on models trained with a dropout of 0.3. Now, the next tables represent precision, recall and f1-score evaluated on holdout data using LSTM and GRUs.

Table 4. Overview of Precision, Recall and F1-score for (a) LSTM and (b) GRUs



```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        10
           1       0.00      0.00      0.00         2
           2       0.26      0.71      0.38         7
           3       0.00      0.00      0.00        10
           4       0.19      0.42      0.26        12
           5       0.00      0.00      0.00         1
           6       0.00      0.00      0.00         4
           7       0.33      0.08      0.13        12
           8       0.00      0.00      0.00         6
           9       0.15      0.36      0.22        11

   micro avg       0.20      0.20      0.20        75
   macro avg       0.09      0.16      0.10        75
weighted avg       0.13      0.20      0.13        75
```

(a)



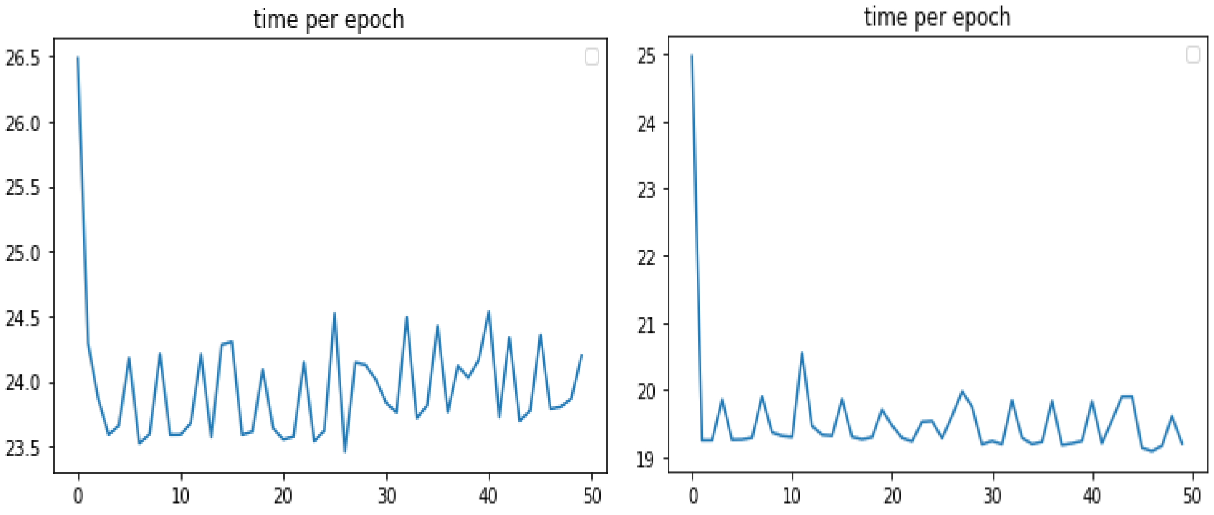
```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        10
           1       0.00      0.00      0.00         2
           2       0.26      0.71      0.38         7
           3       0.00      0.00      0.00        10
           4       0.21      0.33      0.26        12
           5       0.00      0.00      0.00         1
           6       0.00      0.00      0.00         4
           7       0.39      0.58      0.47        12
           8       0.40      0.67      0.50         6
           9       0.22      0.18      0.20        11

   micro avg       0.29      0.29      0.29        75
   macro avg       0.15      0.25      0.18        75
weighted avg       0.19      0.29      0.22        75
```

(b)

In table 4, 0-9 classes represent events in order: 3-pointer success, 3-pointer failure, free-throw success, free-throw failure, layup success, layup failure, other 2-pointer success, other 2-pointer failure, slam dunk success, slam dunk failure.

The results with respect to precision, recall, and f1-score reveal that the performance of the models differs slightly. Since the classes are highly imbalanced, the performance varies in both the models for each class. Therefore, the weighted average precision can be more insightful in order to compare the overall performance of the models. The precision for LSTM is 0.13 and that of the GRUs is 0.19, meaning that GRUs could nearly predict 6 more events accurately than LSTM. Few of the events were neither predicted correctly by LSTM nor GRUs, given the lesser number of instances in training and test dataset. Also, a few of the events are more difficult to recognize than others (Ramanathan et al., 2016). The confusion matrix for both the models is presented in Appendix B.

Further, the runtime efficiency of the models was evaluated which can be analyzed from the plots below in Figure 5.



(a)

(b)

Figure 6: Time per epoch in seconds (a) LSTM (b) GRUs model (dropout rate – 0.3)

Not surprisingly, the training runtime per epoch was lower in case of GRUs, as it has been said to be more computationally simpler and efficient in previous researches as well (Chung et al., 2014; Fu, Zhang, & Li, 2016; Moumen et al., 2018).

6. Discussion

This section discusses the results of the experiments and the implications of each result aspects. Further, it also points out the areas where there is the scope of improvements. Lastly, it discusses the future work that can be the takeaway from this thesis in similar tasks.

To begin with, the research question to be answered in this thesis was "*How do LSTM and GRUs perform in the task of event classification in sports videos?*"

Concerning the task of event classification, a subset of the data was taken from the study done by Ramanathan et al. (2016). The chosen data was then split into train, test and validation data. The preprocessing of the data included the steps for extracting video-clips representing the events from the entire match. Then these video-clips were used to extract the frames in temporal order. The VGG16 architecture was then used to extract features from the 2D frames, which was then stacked together to form a sequence of features to denote a particular event. These sequences of features were used as an input to LSTM and GRUs.

In the experiment, both the models had 3 gated RNN layers, given the complicated task of video classification. Initially, the model network was tested on three different dropout values 0.3, 0.5, 0.9. It was found that both GRUs and LSTM incurred a comparatively lesser loss when the dropout value was 0.3. Also, the accuracy went from 12% to 33% when the dropout was changed from 0.9 to 0.3 in the case of GRUs. But there was no considerable change in LSTM model with respect to varying dropouts. To keep the evaluation consistent, the dropout value was set to 0.3 for both the models and then the metrics were compared.

Taking a closer look into the mean class accuracy, it was found that the model with GRU layers had an accuracy of 4.3% higher than LSTM and 14.47% higher than the minimal defined baseline. LSTM model also learned from the classification task and yielded 10% more accuracy than the baseline. However, training accuracy was higher than validation accuracy, suggesting the overfitting. This might have happened due to improper training with a small dataset. Moreover, the accuracy came with an expense of loss incurred in both the models. Both of the models faced a higher and significant validation loss than training loss, which might be the result of overfitting of data. Due to this, the model found difficulty while converging to minimum errors. It can be assumed that with more number of epochs, the loss in case of GRUs model would be converging better, as it was lowering down in last few epochs. However, LSTM model showed extreme random fluctuations. Higher entropy loss represented the wide divergence between the probability distributions.

Next, since the accuracy metric could still be biased due to the imbalanced dataset used, the precision scores show that the true positives in the case of GRUs were also better as compared to the LSTM. In the model having LSTM layers, the weighted average precision score was 6% lower than the model having GRU layers. Typically talking about the majority class (other 2-pointer failure), GRUs model could detect 5 of the instances correctly out of 12, showing the precision score of 39% and LSTM for the same event scored 33%. While experimenting with the large-scale dataset in the study of Ramanathan et al. (2016), they observed the precision score of 47.1% in classifying the other 2-pointer failure event, and their overall average precision score was 51.6%. They also noticed that the classes with more instances were more accurately classified as compared to the classes with lesser instances. This was seen in the present experiment as well. Further, to evaluate the models in terms of runtime efficiency, LSTM was clearly beaten by the computational efficiency of GRUs, where GRUs took shorter intervals to run an epoch.

Admittedly, there were few points where there is room for improvement which is discussed in the following sub-sections.

6.1 Points of Improvement

Deep learning has enormous quality traits in the vision-based task as mentioned by various researches. But it definitely needs large-scale data for training for better generalizable results and high computational power (Ullah et al., 2018). Due to lesser processing power and time constraints, the data was down-scaled and thus, the results were not comparable to the study done by Ramanathan et al. (2016). Additionally, the data was imbalanced due to which the training was unsatisfactory and contributed almost negligible in the case of minority classes. This issue could have been solved using data augmentation in minority classes. This procedure, if added in preprocessing of data could have resulted in overfitting reduction.

6.2 Future Research

Taking into account the time constraints, the quality of the experiment for event classification was not enhanced using object and player tracking techniques as reflected in a study done by Ramanthan et al. (2016). Future work can impose player and ball tracking techniques before implementing classification training. After enhancing the quality of models with trackers that can inform more about the location of the players, the comparison of LSTM and GRUs can be made. Besides adding this method, data augmentation should be taken care of to deal with imbalanced classes and to generalize the results in similar nature of classification tasks.

7. Conclusion

Regardless of the shortcomings mentioned in the previous section, this research was definitely capable of putting up with its research question. More precisely, gated RNN models trained with multiple layers in the combination of VGG16 for spatial feature extraction performed well as compared to the baseline in order to classify the events in the basketball sports video. The experiment was able to provide an illustration of how the models with deep LSTM and GRU layers can be compared with the use of evaluation metrics of classification. The loss was also found to be converging more in case of GRUs, as compared to that of LSTM.

Consequently, as an answer to the research question, it was noticed that GRUs scores were better than LSTM not only in terms of mean class accuracy and precision, but also in terms of computational runtime efficiency. The findings from this research can be used as a reference for similar video classification tasks using deep learning networks, given that the future work would also enhance the quality of experiment by implementing data augmentation, introducing large-scale dataset and high computational power.

Acknowledgment

The thesis *“An Experimental Comparison of Deep LSTM and GRUs for Event Classification in Sports*” has been written to fulfill the graduation requirements of the Masters in Data Science and Society, at Tilburg University.

The writing of this Master thesis has been a course of development which would not have been in a positive direction without the people accompanying me along the process. I would like to express my sincerest gratitude towards them.

First and foremost, I would like to thank my thesis supervisor Prof. Dr. Eric Postma for providing exceptional guidance throughout the whole process. His feedback and guidance on my work have always been enriched by his vast research experience of deep learning. Our regular meetings in which I had been able to benefit from your valuable suggestions and it also helped me to critically question every step taken thought out the conduct of the research. I would also like to express my sincere gratitude towards Dr. J. Sebastian Olier who guided me with his invariably positive and passionate attitude, he was able to motivate and steer me in the right direction.

The data I used was made publicly available by a research team Ramanathan et al. (2016), I thank them too for their sincere and insight-full project and making the data accessible.

Finally, the thesis concludes a master year at Tilburg University filled with a lot of worthy highlights. I have been able to meet interesting people from all over the world who shared their thoughts and experiences with me and who have helped me grow personally, academically and professionally. Aakash Wadhwa, Priyesh Chhabra, and Rahul deserve an extra mentioning because of their indefinitely extended efforts to help me and be true moral support throughout.

Most importantly, I want to thank my family. Without you, I would not be who and where I am now.

I hope you enjoyed your reading.

Kajal Chandani

Tilburg, May 2019

References

Acharya, D., Yan, W., & Khoshelham, K. (2018). Real-time image-based parking occupancy detection using deep learning. In Research@ Locate (pp. 33-40).

Chang, W. Y. (2019). Research on sports video image based on fuzzy algorithms. *Journal of Visual Communication and Image Representation*, *61*, 105-111.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2625-2634).

Fernando, B., & Gould, S. (2016, June). Learning end-to-end video classification with rank-pooling. In *International Conference on Machine Learning* (pp. 1187-1196).

Fu, R., Zhang, Z., & Li, L. (2016, November). Using LSTM and GRU neural network methods for traffic flow prediction. In *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)* (pp. 324-328). IEEE.

Géron, A. (2017). Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems. " O'Reilly Media, Inc.".

Ghewari, R. S. (2017). Action Recognition from Videos using Deep Neural Networks (Doctoral dissertation, UC San Diego).

Graves, A., Mohamed, A. R., & Hinton, G. (2013, May). Speech recognition with deep recurrent neural networks. In 2013 IEEE international conference on acoustics, speech and signal processing (pp. 6645-6649). IEEE.

Graves, A., Fernández, S., & Schmidhuber, J. (2005, September). Bidirectional LSTM networks for improved phoneme classification and recognition. In *International Conference on Artificial Neural Networks* (pp. 799-804). Springer, Berlin, Heidelberg.

Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2015). Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

Huang, S., Li, X., Zhang, Z., Wu, F., & Han, J. (2019). User-Ranking Video Summarization with Multi-Stage Spatio–Temporal Representation. *IEEE Transactions on Image Processing*, *28*(6), 2654-2664.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

Karlsson, D. (2017). Classifying sport videos with deep neural networks.

Karpathy, A. (2015). The unreasonable effectiveness of recurrent neural networks. Andrej Karpathy blog, 21.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 1725-1732).

Khandelwal, S., Lecouteux, B., & Besacier, L. (2016). *Comparing Gru And Lstm For Automatic Speech Recognition* (Doctoral dissertation, LIG).

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

Liang, M., Hu, X., & Zhang, B. (2015). Convolutional neural networks with intra-layer recurrent connections for scene labeling. In Advances in Neural Information Processing Systems (pp. 937-945).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436.

Madani, A., Arnaout, R., Mofrad, M., & Arnaout, R. (2018). Fast and accurate view classification of echocardiograms using deep learning. NPJ digital medicine, 1(1), 6.

Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z., & Yuille, A. (2014). Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*.

Miech, A., Laptev, I., & Sivic, J. (2017). Learnable pooling with context gating for video classification. arXiv preprint arXiv:1706.06905.

Moumen, R. A., Chiheb, R., Faizi, R., & El Afia, A. (2018). Evaluation of Gated Recurrent Unit in Arabic Diacritization. *International Journal Of Advanced Computer Science And Applications*, *9*(11), 360-364.

Mookiah, M. R. K., Acharya, U. R., Chua, C. K., Lim, C. M., Ng, E. Y. K., & Laude, A. (2013). Computer-aided diagnosis of diabetic retinopathy: A review. Computers in biology and medicine, 43(12), 2136-2155.

Nepal, S., Srinivasan, U., & Reynolds, G. (2001, October). Automatic detection of'Goal'segments in basketball videos. In *Proceedings of the ninth ACM international conference on Multimedia* (pp. 261-269). ACM.

Ning, C. (2019). Design and research of motion video image analysis system in sports training. *Multimedia Tools and Applications*, 1-19.

Petscharnig, S., & Schöffmann, K. (2018). Learning laparoscopic video shot classification for gynecological surgery. Multimedia Tools and Applications, 77(7), 8061-8079.

Pham, V., Bluche, T., Kermorvant, C., & Louradour, J. (2014, September). Dropout improves recurrent neural networks for handwriting recognition. In 2014 14th International Conference on Frontiers in Handwriting Recognition (pp. 285-290). IEEE.

Ramanathan, V., Huang, J., Abu-El-Haija, S., Gorban, A., Murphy, K., & Fei-Fei, L. (2016). Detecting events and key actors in multi-person videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3043-3053).

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A.,Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet large scale visual recognition challenge. CoRR,abs/1409.0575, 2014.

Rodriguez, Mikel D, Ahmed, Javed, and Shah, Mubarak. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In CVPR,2008.

Saur, D. D., Tan, Y. P., Kulkarni, S. R., & Ramadge, P. J. (1997, January). Automated analysis and annotation of basketball video. In *Storage and Retrieval for Image and Video Databases V* (Vol. 3022, pp. 176-188). International Society for Optics and Photonics.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Staudemeyer, R. C., & Omlin, C. W. (2013, October). Evaluating performance of long short-term memory recurrent neural networks on intrusion detection data. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference* (pp. 218-224). ACM.

Tsai, Y. H. H., Divvala, S., Morency, L. P., Salakhutdinov, R., & Farhadi, A. (2019). Video Relationship Reasoning using Gated Spatio-Temporal Energy Graph. *arXiv preprint arXiv:1903.10547*.

Tsunoda, T., Komori, Y., Matsugu, M., & Harada, T. (2017). Football action recognition using hierarchical LSTM. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 99-107).

Ullah, A., Ahmad, J., Muhammad, K., Sajjad, M., & Baik, S. W. (2018). Action recognition in video sequences using deep bi-directional LSTM with CNN features. *IEEE Access*, *6*, 1155-1166.

Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R., & Saenko, K. (2014). Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*.

Wang, H., Zhao, G., & Yuan, J. (2014). Visual pattern discovery in image and video data: a brief survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *4*(1), 24-37.

Wang, M., Chu, B., Liu, Q., & Zhou, X. (2017, August). YNUDLG at SemEval-2017 Task 4: A GRU-SVM Model for Sentiment Classification and Quantification in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (pp. 713-717).

Wang, S., Wang, Y., Tang, J., Shu, K., Ranganath, S., & Liu, H. (2017, April). What your images reveal: Exploiting visual contents for point-of-interest recommendation. In Proceedings of the 26th International Conference on World Wide Web (pp. 391-400). International World Wide Web Conferences Steering Committee.

Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., & Toderici, G. (2015). Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4694-4702).

Yu, J., Lei, A., & Hu, Y. (2019, January). Soccer Video Event Detection Based on Deep Learning. In International Conference on Multimedia Modeling (pp. 377-389). Springer, Cham.

Zhang, Y., Xu, C., Rui, Y., Wang, J., & Lu, H. (2007, July). Semantic event extraction from basketball games using multi-modal analysis. In *2007 IEEE International Conference on Multimedia and Expo* (pp. 2190-2193). IEEE.

Zhu, L., Xu, Z., & Yang, Y. (2017). Bidirectional multirate reconstruction for temporal modeling in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2653-2662).

Appendix A: Codes and their purposes

- downloading_videos.py: code to download and save YouTube videos
- extracting_subclips.py: extracting the YouTube video for given start and end timestamps
- video2frames.py: converting video-clips into frames
- splitting_data.py: partition the data in train, validation, and test sets
- data_proc.py: extracting features using VGG16
- main.py: stacking features from frames to form a sequence of input, the definition of model trainings (LSTM and GRU), finally, evaluation and plot generation
- Sample_data: containing a few instances to give an outlook of data-structures.

GitHub link to access the content: https://github.com/KajalChandani/Master-Thesis-

Appendix B: Confusion Matrix for event classification

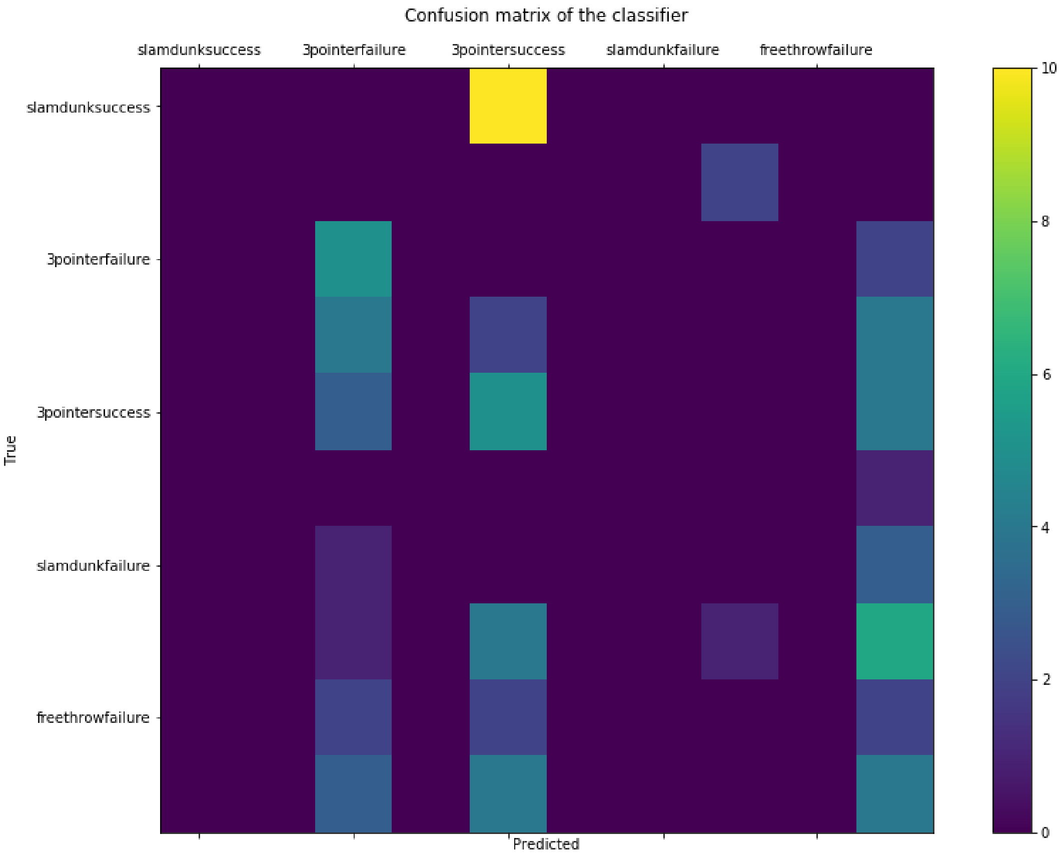


Figure 6: Confusion matrix for LSTM (dropout rate – 0.3)

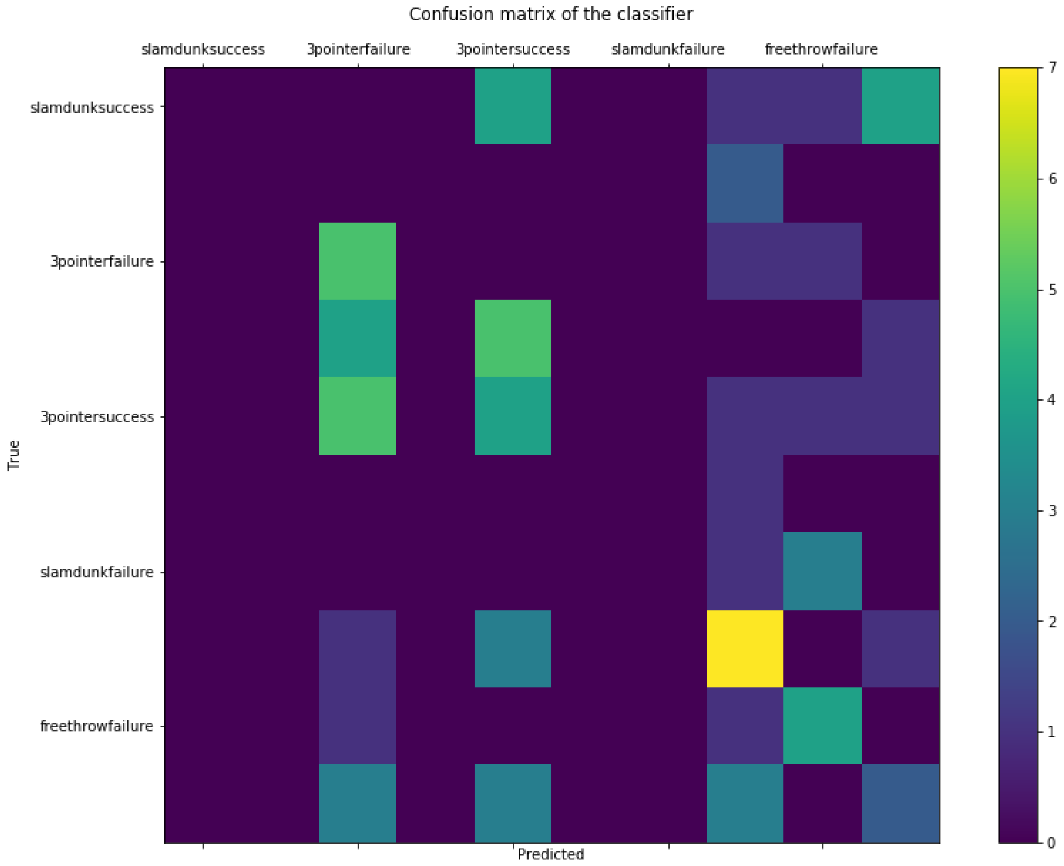


Figure 7: Confusion matrix for GRU (dropout rate – 0.3)