ACTIVITY RECOGNITION TO MIMIC HUMAN PERCEPTION

by

ALANKRIT GUPTA

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2020

To

LIFE

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervising professor Dr. Manfred Huber for constantly advising and encouraging me. His invaluable advice, deep expertise in various computer science sub-domains and huge breadth in computer science as a whole has proved to be extremely helpful during the entire course of my graduate studies. I wish to thank my committee members Dr. Vassilis and Dr. Kamangar for their interest in my research and for taking time to be in my committee.

I would also like to extend my appreciation to Dr. Huber for giving me a stress-free environment for continuing my research on many topics I have been interested in.

I am grateful to all the professors who taught me during the years I spent as a graduate student. I would like to thank Dr. Kamangar whose neural network class inspired me to study neural networks using online courses.

Finally, I would like to express my deep gratitude to my parents who have encouraged me and also sponsored my graduate studies.

Aug 10, 2020

ABSTRACT

ACTIVITY RECOGNITION TO MIMIC HUMAN PERCEPTION

ALANKRIT GUPTA, M.S. Computer Science

The University of Texas at Arlington, 2020

Supervising Professor: Dr. Manfred Huber

The recognition of activities from video is a capability that is important for a wide range of applications, ranging from basic scene understanding to the successful prediction of behavior in autonomous vehicle applications. At this time, human capabilities in this task by far outperform computer applications and thus the idea to mimic human perception should be promising. In this thesis we are proposing an architecture that processes videos to extract important action instances that describe the essential behaviors contained in any video and help us map the information from the video to a machine-understandable form. This is an important research area, as it could help us interpret the surrounding environment for the visually impaired, detect and characterize human behavior for autonomous vehicles, as well as enhance security at some of the most vulnerable places by identifying suspicious behavior. All of this illustrates the vast range of possibilities to this technology. The architecture proposed here is divided into three major sub-modules, namely: i) Localization; ii) Action Detection; iii) Description mapping. In this thesis, all the submodules are introduced and their interaction and operation is described before the

action detection module is implemented and its performance is demonstrated. In addition, the thesis will describe how we could use transfer learning to combine all the proposed specialized components to mimic human perception.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iv

ABSTRACT v

LIST OF ILLUSTRATIONS viii

Chapter Page

1. Introduction 1

2. Related Work 3

3. Methodologies 8

3.1 Localization Network 9

3.1.1 Temporal Segmentation 10

3.1.2 Proposal 11

3.2 Classification Network 12

3.3 Description Mapping 14

4. Classification Model Implementation 16

4.1 Experiments and Results 17

4.2 Conclusion 24

5. Conclusion 26

6. References 27

LIST OF ILLUSTRATIONS

Figure Page

3.1 Proposed Architecture 8
3.2 Proposed Localization Module 10
3.3 Proposed Classification Module 12
3.4 Proposed Description Mapping Module 14
4.1 Experiment 1 Architecture 18
4.2 Experiment 1 Classification Model accuracy & top-5 19
4.3 Experiment 2 Architecture 20
4.4 Experiment 2 Classification Model accuracy & top-5 21
4.5 Experiment 3 Architecture 22
4.6 Experiment 3 Classification Model accuracy & top-5 23
4.7 Classification Model Architecture 24

CHAPTER 1

Introduction

In recent years, the field of Artificial Intelligence (AI) and in particular of Machine Learning has made significant advances, allowing for increasingly more complex tasks to be addressed. This has opened up many new capabilities and advanced the drive of some researchers towards the ability to mimic all the possible human components on a machine. With the recent advances in Neural Networks (NN), which has led to some significant advancements where these networks are used to predict stocks, classify images, build a predictive model to learn from trends and for many other tasks.

All this progress has revived a drive towards attempts to build a system capable of implementing systems that can address a wide range of tasks a human is capable of achieving. Keeping this in mind the work in this thesis is driven towards proposing a technique focused to mimic human perception. Currently, human perception is one of the most researched areas, with the idea for the machine to understand and perceive an environment as we do. As perception is mostly considered relative, the focus is mostly on being able to understand some of the most important events in time and be able to understand the correlation between them.

The drive to mimic human perception also often underlies efforts to be able to build an assistive device focused on helping visually impaired people with some of the most basic daily tasks. This could help provide life-changing assistance to them, while also giving them an outlook of the world around them. The possibility of this technology does not just end here, as work in this direction could also pave the way to secure some of the

most vulnerable locations like airports, religious venues, secured official government buildings, and many more. As we could use the basic components of this architecture to predict a possible localized attack.

The work proposed here to mimic human perception in the area of video interpretation is divided into three specialized sub-components. The first specialized component is Localization, with the focus on localizing potential activity instances in time, from untrimmed video input. This is an important aspect as it helps identify multiple potential action instances taking place in time since an event taking place over a few minutes or maybe more might have multiple correlated events taking place. The second component is Classification, which helps identify a potential activity taking place in the proposed action instance. This plays a crucial role as it provides a summary of a potential action instance which could help drive focused information extraction to map various events in time. The third component is Description mapping, which focuses on helping map various events in time to their description. The proposed architecture uses the feature components extracted during classification to establish the relation between different proposals over time.

CHAPTER 2

Related Work

As the work in this thesis is mainly focused with localization, action recognition, and classification, this related work will focus on the most closely related work in these areas and show how these sub-components have evolved over time and how the approach presented in this thesis, while learning from these approaches, proposes different outlooks to them. This section will also discuss how we can work to combine different techniques to help learn over different modules.

ACTIVITY RECOGNITION. Earlier work for the task of activity recognition has focused on using predefined concepts, such as in SVO [1], to identify elements in the frame to focus on to retrieve information. As the move towards more object-centrically focused detection took place, techniques were developed that focused information extraction around objects of interest while still using hand-crafted features [12] [13]. With the advances in computing and machine learning, most of the most recent work is driven by deep learning features to understand the activity. This is crucial as it gets rid of building specialized components, which potentially had an issue with transfer learning.

LOCALIZATION. To localize events in time, a common approach has been to use a sliding window approach which helps to extract events in time with varied lengths. This approach defines different sized windows to look at the frames together, an approach to possibly understand the relation between different frames when seen together [1]. Though the sliding window approach might help localize every possible event of varied length, it seems to come with quite an overhead which might be unnecessary. In this work, we are

proposing a variant of the sliding window, in which we sub-divide video into smaller 16 frame components, referred to as 1 sub-mod, and taking multiple sub-mods together to a specific limit. This two-level mechanism is aimed at helping in detecting localized events with a lower overhead.

For example, taking sub-mods at a stride rate of 1, 2, 4, 8 might help extract relevant information for increasingly long activities. The reason for choosing a smaller window of 16 frames is that it usually provides sufficient information regarding a local change taking place within the frames. The pick of 16 frames here is based on experiences form a significant amount of prior research experiments [4] [5]. To illustrate this, consider an average video has a frame rate of anywhere between 25-30 frames per second (FPS). Considering we take 16 frames for a sub-mod, taking stride at the rate 1, 2, 4, 8 will help us analyze 16, 32, 64, and 128 frames together, which roughly translates to >.5 sec, >1 sec, > 2 secs, > 4 seconds. Though this number might seem small, it helps to locate an important activity taking place, as the action change should be summoned in this period. Even though a normal activity might extend across longer periods, this could help us to identify localized changes in events, including the beginnings and ends of the longer activity. With the combination of localizing multiple smaller proposals together, the overlapping proposal with the same class could be identified as a single event.

SPATIO-TEMPORAL ACTION LOCALIZATION. Recently there has been a lot of interest in Spatio-temporal aspects of action localization [2]. This is a crucial component as it helps understand important spatial components of every frame in a segment. This is usually referred to as visual encoding, as it helps in extracting important visual component features. Usually, a pre-trained CNN is used to extract features of an individual frame, or

a pre-trained C3D network is used to extract features of a segment consisting of multiple frames [6] [8]. This vastly helps in directing spatial focus towards the important features in a frame, as these networks are extremely deep and are usually trained on huge image-net datasets [14]. Considering the other aspect of spatio-temporal localization is temporal localization, also referred to as sequence encoding. This helps us to understand the relation between different frames and how the change from one frame to another helps understanding the activity taking place. For temporal localization, LSTM's [7] are frequently used to learn the relations within the sequences of frames. Spatio-temporal localization is frequently used for recognition and classification from video, to help understand the relationships within a frame as well as in between different frames [15].

SST: SINGLE-STREAM TEMPORAL ACTION PROPOSAL [4]. The approach for Single-Stream temporal (SST) action detection proposes an alternative to the sliding window approach that is used to describe the action in a given video. The sliding window approach can be computationally expensive as a result of requiring multiple passes over the same video with different temporal scales. In contrast, SST is able to perform its processing in a single pass through the window. To achieve this they use a 3D Convolutional (C3D) [6][8] network for video input, and train it to effectively capture visual and motion information at a small temporal resolution. In order to accumulate evidence over time to allow the model to be able to aggregate information so as to determine if an action has taken place while ignoring the irrelevant background, this model uses recurrent network layers. Since the model needs to process the video in a single pass, the recurrent models here need to unroll over the entire input testing video. GRU based architectures were found to provide better performance and are thus used here. At each

time step, the model outputs confidence scores of multiple proposals. These scores themselves are here learned by fully connected layers. The design approach to include recurrent networks to fully unroll over the entire video sequence acts as a key property that enables the model to operate without using overlapping sliding windows. In this work, the authors also observed that the hidden states in the recurrent networks tend to saturate when running over many steps resulting in overconfident results. This approach is functionally very useful but uploading the whole pre-trained C3D Network on the GPU memory even after getting rid of some of the top layers proves to be a huge limitation.

DAP's: DEEP ACTION PROPOSAL FOR ACTION UNDERSTANDING [5]. Focusing on the success of object proposals in object understanding in images Escorcia et al introduced a new approach named Deep Action Proposals (DAP's) as an efficient technique to generate temporal action proposals from videos. The proposed architecture retrieves fidelity proposals with lower computational costs. In order to move forward with high-level analysis of long untrimmed videos, they suggested to put the development of action proposal at the forefront of human activity understanding research. The new proposed approach is trained to output temporal locations and scales to a fixed number of proposals. The model generates proposals at multiple temporal scales with a single pass, including for previously unseen actions. The approach is computationally efficient and runs at 134 FPS. They reduce the number of evaluated windows by encoding the video in a sequence of visual descriptors. For temporal action proposals they create a hierarchy of fragments by hierarchical clustering, based on semantic visual similarity of continuous frames. For the implementation DAP uses a pretrained C3D reference model as a visual encoder. The DAP network reduces the dimensionality of the representation from the

second fully connected layer from 4096 to 500 using PCA. DAP's action classifier encodes features learned by conv-net using VLAD [16]. To measure the quality of temporal proposals, they use average recall. In this thesis we will be using a modification of DAP to help us propose if a video instance contains a possible action or not.

TEMPORAL ACTION LOCALIZATION IN UNTRIMMED VIDEOS VIA MULTISTAGE CNN [3]. In most work on video processing the datasets used fall into one of two types: i) Data that has video level category labels but no temporal annotations (weekly supervised); or ii) Data where temporal boundaries have been annotated in untrimmed videos. Both of these different types of datasets could help us build networsk focused on different segments of the proposed module. In [3] the authors employ multi-stage segment extraction, a windowed approach, to extract frame segments at different lengths with the focus on classifying them as being either an action or a background. The action segments are then sent to a classification network, together with an equal distribution of background segments, to help learn classes the actions belong to. They later have their own version of a localization network with the focus on increasing the score of proposal with high Intersection over union (IoU) [9] with the ground truth. This is further complemented with Non-Maximum Suppression (NMS) [10] to get rid of proposals with a higher IoU access to the ground truth and higher score proposals. While this seems to be a good approach, we believe they could extract more information from the discarded proposals. Here we propose to use them in our later stages to build a proposal segment with longer length.

CHAPTER 3

Methodology

The proposed architecture to mimic human perception consists of three specialized components. This thesis is proposing the architecture for the whole system, introducing each of them with a brief description, while focusing on the classification component in more detail and implementing a prototype version of it. The work proposing the architecture is mainly focused on how we can incorporate differently trained specialized components to help mimic human perception. Since the requirements for each component might vary, this work focuses on how we can use transfer learning as well as on the benefit of one specialized unit to improve the utility of another.

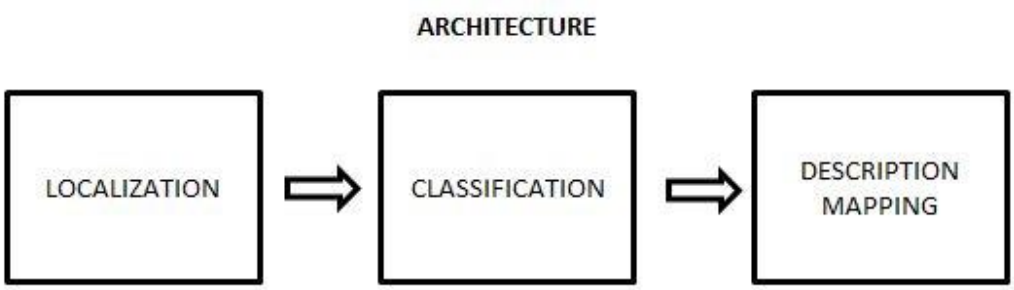

Figure 3.1: Proposed Architecture

The specialized components comprising the proposed architecture are:

1. Localization Network: The goal of this component is to temporally localize activities within the untrimmed video.
2. Classification Network: This component's objective is to identify (classify) the activity in the localized segment.
3. Description mapping: The goal of this component is to map the activity sequence to a description of the video content.

3.1 Localization Network

As the main focus of our proposed architecture is to mimic human perception, one aspect of the perception that we need to focus on right away is that a single scene depiction or a video might have more than one action instance, which further might be of varied lengths. This is the major concern or challenge moving forward with respect to the localization module. The localization we are here referring to is mostly concerned with localizing an activity or action instance in time. For the proposed architecture we further subdivided this module into two smaller but complimenting sub-components.

The components as shown in Figure~3.2 are:

1. Temporal Segmentation
2. Proposal

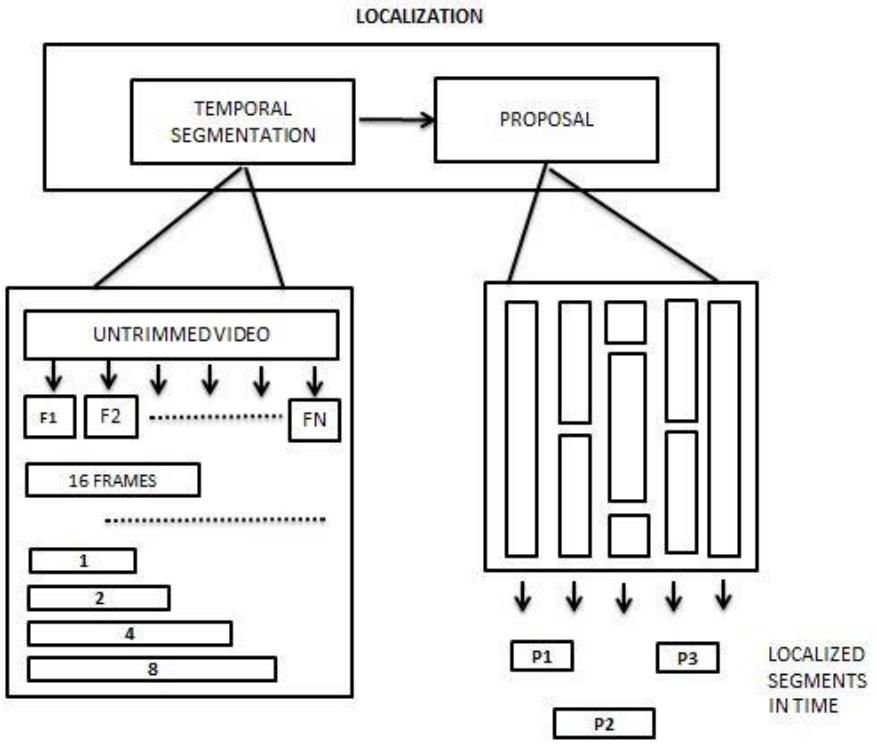

Figure 3.2: Proposed Localization Module

3.1.1 Temporal Segmentation

The challenge to identify localized action instances in time is generally computationally expensive due to the enormous number of possible permutations and combinations of all possibly consecutive frames that could correspond to a possible action segment proposal. However, this is a necessary evil when concerned with being able to detect events in a long and untrimmed video or scene depiction. With the proposed architecture we are trying to provide a relatively viable solution that could help extract proposals of varied length while also considering the computational overhead.

The proposed temporal segmentation module is mostly aimed at long and untrimmed video sequences. To help with localized segment retrieval, the first step is to segment the entire video into individual frames. This will help to form smaller segmented sequences and also be useful when retrieving features for each frame in a segmented proposed sequence.

The approach then then combines frames into sets of 16 frames (1 sub_mod) with a skip of 16. This leads to a division of the length of a whole video sequence into *t* localized sub_mods, where *t* = (length of video)/16. These sub_mods are then combined with strides 1, 2, 4, 8, which means that we combine the adjacent *n* sub_mods to build a sequence length of *n·16* frames for $n \in \{1, 2, 4, 8\}$, capturing sequences of different lengths.

This extraction process helps us find events in longer sequences while avoiding to generate a large overhead to accommodate every possible combination of adjacent frames. We will later in the description of our novel classification network define how this proposed method for temporal segmentation can help us achieve better results with lower overhead when compared with a moving window approach.

3.1.2 Proposal

After extracting segments from the untrimmed video sequence, we use a variant of DAP's (Dense Action proposal) [5] to propose probable segment sequences that might contain action instances. DAP's provide us with a confidence score for each proposal, allowing the system to select the *k* proposals with the highest scores.

DAP's is trained using a procedure in which it proposes a segment S_i with a probability C_i. The goal of this procedure is that the action proposed by the model should match the location of actions in the sequence.

These localized events/actions in time help us to extract proposals localized with high action likelihood. The output from this module will act as an input to our classification network

3.2 Classification Network

After retrieving the possible action segment sequences from the localization network, we use those proposed segments as input to the classification network. Since these proposed segments are trimmed using the likelihood of an action taking place from the start of the proposed sequence to the end of it, this acts as a perfect input to our classification network.

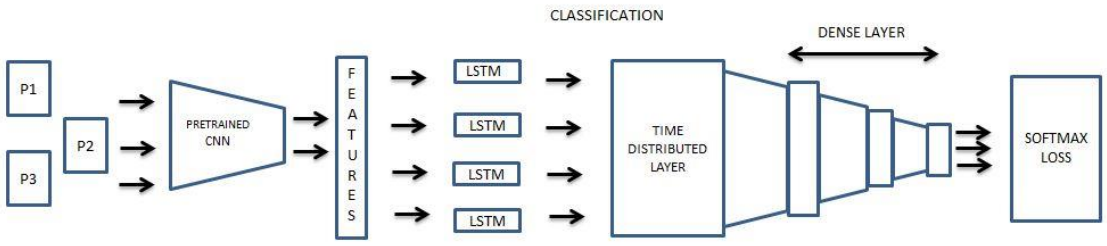

Figure 3.3: Proposed Classification Module

The classification architecture uses the concepts of transfer learning to extract visual encodings for the proposed segmented sequences. We are using the Inception V3 pre-

trained CNN model on ImageNet to extract spatial features for each frame in the given localized sequence [17]. This avoids training this component from scratch and takes advantage of the significantly larger image data set used to train the ImageNet architecture. Starting with this we can more efficiently build our classification network.

The extracted features act as an input to LSTMs at varied time steps since each proposed action sequence is of varied length through sub_mod striding at 1, 2, 4, and 8, corresponding to the segment lengths generated by the temporal segmentation component. The LSTM sequence here acts as a sequence encoder as it extracts information over time with each frame feature in the sequence representing a single time frame.

The output of the LSTM is preserved for every time sequence, serving as the input to a Time Distributed Layer which considers the entire sequence of actions over time to preserve all vital information, as is is common for LSTMs to forget older input over long sequences. This technique helps us preserve the information over long sequences of input proposals.

The output extracted from the time distributed layer is then flattened and fed into a dense network which is connected to a SoftMax layer to predict the action class of the given proposal.

As mentioned earlier in the localization module, the proposed method for localization can help achieve better results with less overhead compared to a moving window approach. This is achieved through a combining stage after the classification network. After classifying each proposed video of smaller segments of 16, 32, 64, or 128 frames, we run a combining task. The purpose of this is to determine if any number of nearby action proposals represent the same class. If this is the case, these proposals will be

combined to represent a single action sequence of longer length, resulting in many cases in segments that are significantly longer than our smaller window lengths. This technique acts as a crucial element to reduce the computational overhead while also extracting action sequences of longer duration.

3.3 Description Mapping

The description mapping uses the information provided by both our previous specialized components, i.e. the sequence of action proposals to map these to descriptions for the entire video sequence. After running the combining task that combines the nearby proposal sequences, we pass those combined proposal segments again through the classification network to extract hidden states of those elements. These hidden states are extracted from the last fully connected layer of the classification network just before the SoftMax layer and provide the input to the Description Mapping component as shown in Figure 3.4.

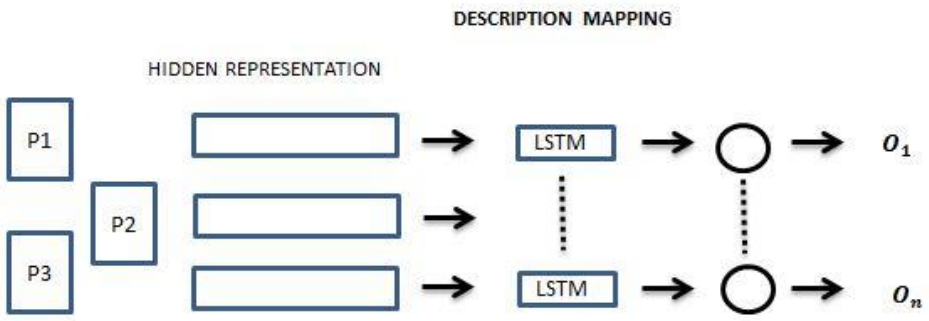

Figure 3.4: Proposed Description Mapping Model

Since we have all the information from the hidden layer before the SoftMax layer of the classification network, we feed those hidden representations through LSTMs [11] to map them to descriptions. To capture the relationships between different proposals which are part of the same video, we feed these hidden representations of the proposals through the Description Mapping network based on the order at which these action proposals took place in time.

Some of the datasets that can be used to train these specialized components in a supervised method are the MPII-MD movie dataset and the Activity Net Caption dataset [18], as both of them are vast datasets with localized captions. This will play a crucial role in order to map these hidden representations to their descriptions.

CHAPTER 4

Classification Model Implementation

For our classification network implementation, we have used UCF101 [19] as our primary dataset. To make the training process easier, we are working here with some data preprocessing steps.

One of the most important components while handling this dataset is to consider an efficient and unbiased way of splitting test and training set initially as the dataset contains videos of action sequences belonging to the same group. In particular, in this dataset separate videos were recorded but with the same actors and the same camera settings. Considering this is important since, if videos belonging to the same group are in both the train and test set, this could falsely increase the validation accuracy.

The documentation for UCF101 comes with a train/test split guideline which we are following here. This splits the data into a training set with more than 8596 videos, and a test set with more than 3418 videos. We are using tests as our validation set to perform an early stoppage if validation loss does no longer decrease after a while.

Using these sets we are subdividing each video into individual frames, helping while retrieving a certain sequence length with respect to each video and also helping during feature extraction.

After extracting frames from each video, we form sequences of 40 frames for each video and only extract frames from videos of length less than 300 frames. Taking such steps helps us extract a frame sequence for every 7.5 seconds of video, which should help the network to understand the change in action taking place.

These frame sequences are then passed into the Inception V3 pre-trained CNN on ImageNet, to extract a spatial feature encoding. This process is run separately from our classification network to have all the features available ahead of time and not to have to re-run the pre-trained Inception network. This helps in faster training by avoiding redundant computations as we are not refining the Inception network's weights.

Features are extracted for each frame using a feature vector of dimension 2048. These features are fed into our LSTM unit for the 40-time steps of the temporal segments described above. For the LSTM unit, we return all the hidden time step outputs instead of just the final learned LSTM layer output so that we have all the information preserved with regards to each time step.

4.1 Experiments and Results

In the experiments presented here we have worked with early termination and thus a relatively small number of iterations of the architecture to avoid overfitting, as some architectural changes led to accuracies on the training dataset of close to 100% when trained for a long time where additional training led to no further improvements for our validation set. In some cases the longer training on the train dataset even led to a decline in the validation accuracy.

We are optimizing using Adam optimizer with a learning rate of 0.00005 and a decay at the rate of 0.000001, with RELU activation in each layer. The experiments use cross-entropy as a loss function and accuracy and top-k categorical accuracy as our evaluation

metric. We have used early stoppage at 20 epochs, i.e. if the validation loss does not increase after 20 iterations, the model will stop the training process.

A number of different architectures where tested and a number of additional regularization terms and features were included to minimize overfitting further in addition to the early stopping criterion described.

EXPERIMENT 1

The initial experiment architecture uses the proposed architecture components consisting of an LSTM layer, two Time Distributed layers, fully a dense layer and a SoftMax layer. In addition, it includes a Dropout layer for additional regularization between the LSTM and Time Distributed layers as well as between the Dense and the SoftMax layer. The goal here was to further reduce the potential for overfitting. The complete used architecture is as follows:

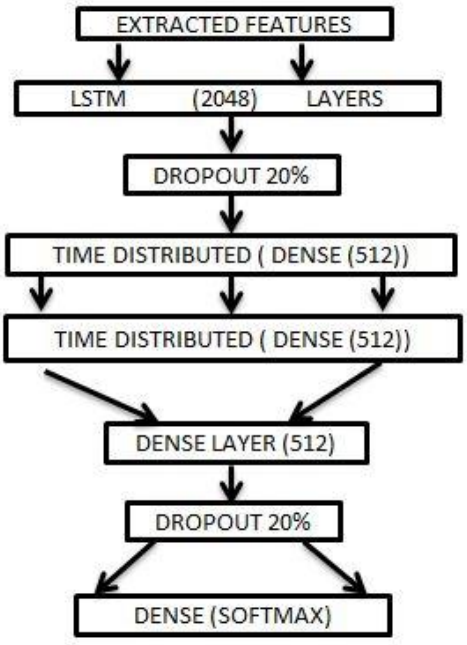

Figure 4.1: Experiment 1 Architecture

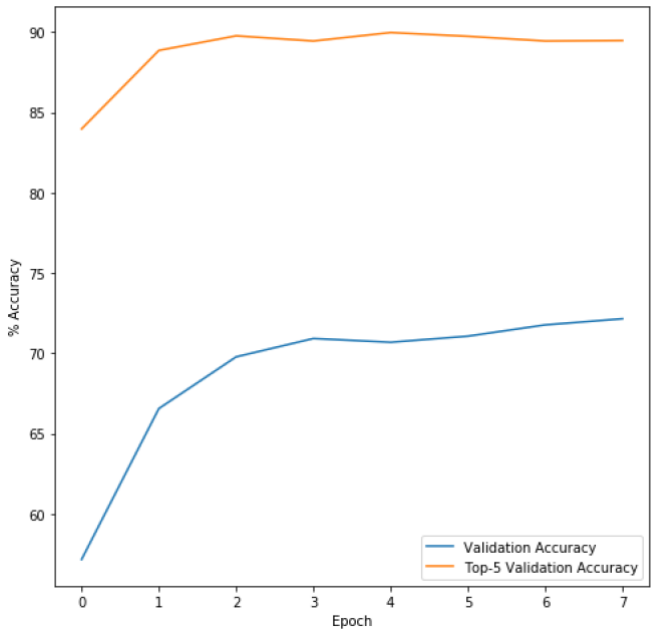

Figure 4.2: Experiment 1 Classification Accuracy and Top-5 Accuracy

This architecture led the model to a training accuracy of 99.5 % just within 8 epochs. The validation accuracy, on the other hand, peaked at 72.14% accuracy. This indicates significant overfitting as to the best results for the model are achieved in a short span of 8 epochs. The top-5 categorical accuracy for this model peaked at 89.96%.

EXPERIMENT 2

To address some of the overfitting observed in the previous architecture, a second experimental architecture was built in which a higher dropout rate was combined with an additional dense layer, resulting in the following network architecture;

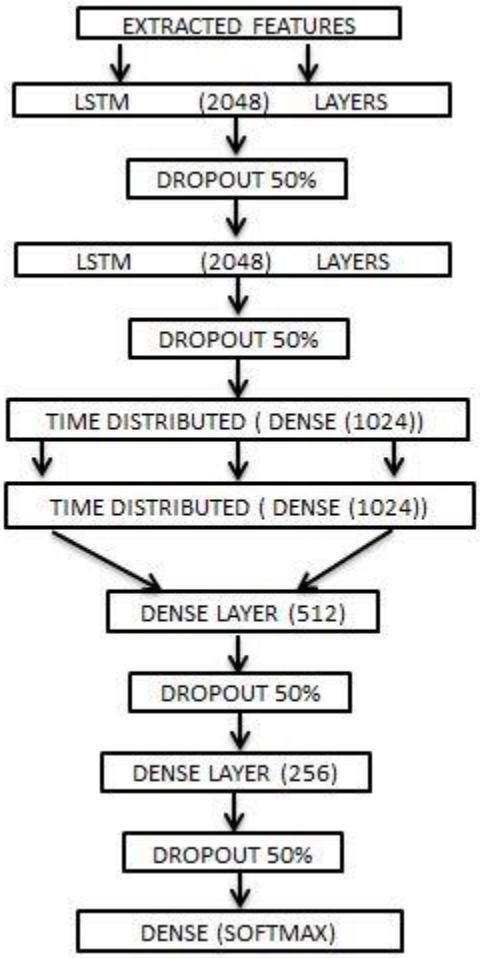

Figure 4.3: Experiment 2 Architecture

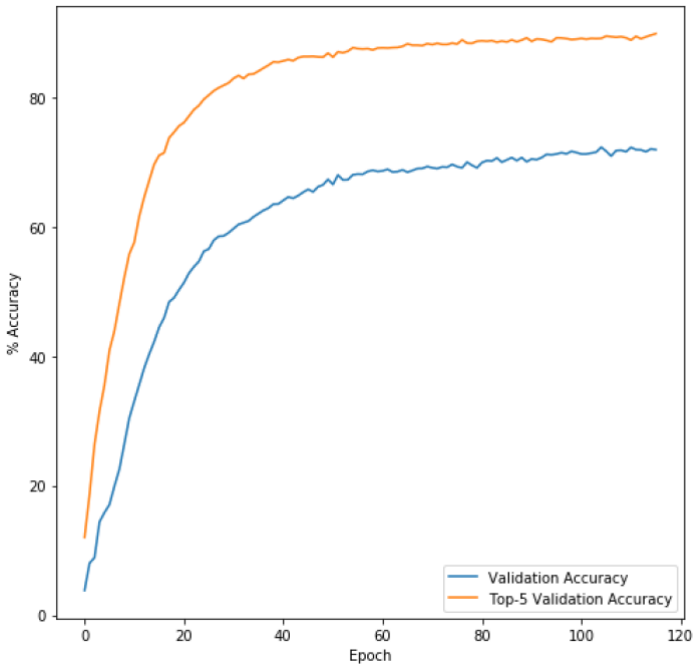

Figure 4.4: Experiment 2 Classification Accuracy and Top-5 Accuracy

In this architecture the model's training accuracy peaked at 90.8 % after 116 epochs. On the other hand, the validation accuracy peaked at 72.32% accuracy. This indicates a significant decrease in terms of overfitting with a small accuracy improvement on the validation set. The top-5 categorical accuracy peaked at 89.87%, which is not significantly different from the previous architecture.

EXPERIMENT 3

To further test whether overfitting could be further reduced, a third architecture was built that reduced the number of dense layers back to 1 while maintaining the higher dropout rate. The architecture for this model is as follows:

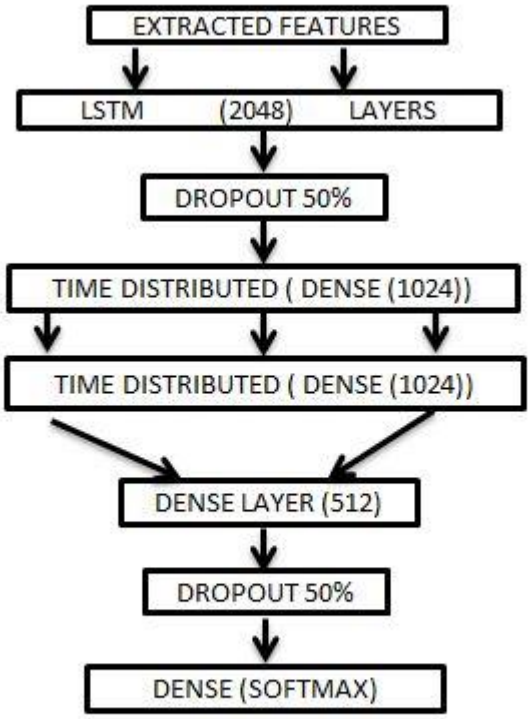

Figure 4.5: Experiment 3 Architecture

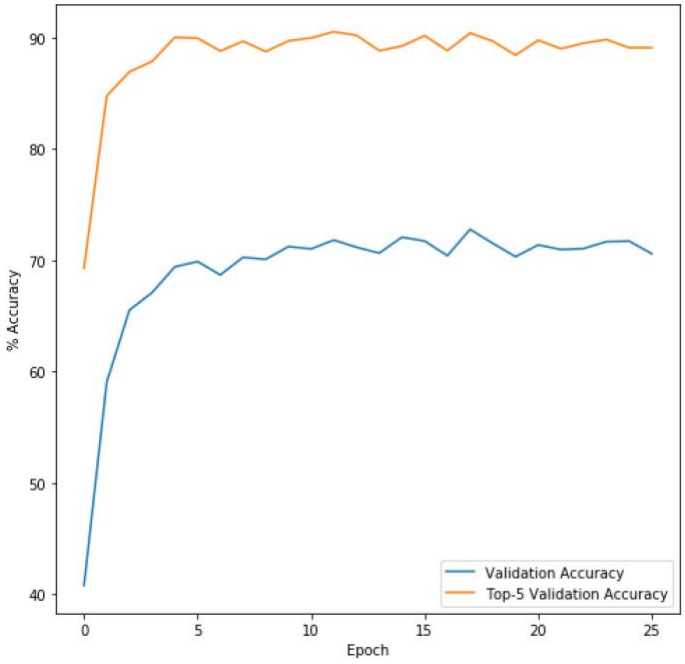

Figure 4.6: Experiment 3 Classification Accuracy and Top-5 Accuracy

This architecture again displayed a higher level of overfitting. The model's training accuracy achieved 99 %, while the validation accuracy peaked at 72.79% accuracy, but soon declined to 70.5% due to additional overfitting. This demonstrates the abovementioned effect of a decline in validation accuracy due to overfitting in later stages of training.

4.2 Conclusion

Within the architectures evaluated here, Architecture 2 as shown in Figure 4.3 proved to be the best model. As reiterated in Figure 4.4, this model showed growth in most of the training process without any decline for a longer period of time, indicating a more stable model than the other two architectures. Thus, we propose it as our model architecture with its details shown again in Figure 4.7.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 40, 2048)          33562624
_________________________________________________________________
lstm_1 (LSTM)                (None, 40, 2048)          33562624
_________________________________________________________________
time_distributed (TimeDistri (None, 40, 1024)          2098176
_________________________________________________________________
time_distributed_1 (TimeDist (None, 40, 1024)          1049600
_________________________________________________________________
flatten (Flatten)            (None, 40960)             0
_________________________________________________________________
dense_2 (Dense)              (None, 512)               20972032
_________________________________________________________________
dropout (Dropout)            (None, 512)               0
_________________________________________________________________
dense_3 (Dense)              (None, 256)               131328
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_4 (Dense)              (None, 101)               25957
=================================================================
Total params: 91,402,341
Trainable params: 91,402,341
Non-trainable params: 0
```

Figure 4.7: Classification Model Architecture

As indicated, our classification network achieved a validation accuracy of 72.32% for top-1 proposal and of 89% for top-5 proposals. As indicated, we chose this architecture, due to it showing the least amount of overfitting model and the most consistent increase in both training accuracy and validation accuracy during training. Though these results are well below the state-of-the-art results, the proposed network is significantly lower

complexity. Moreover, since we did not use the pre-trained networks at the same time as classification, the weights of these network components were not tailored with respect to the given problem and architecture.

CHAPTER 5

Conclusion

This thesis proposed an architecture to mimic human perception and implemented the classification component which addresses human activity recognition. We proposed an architecture consisting of different specialized components that together should increase the utility of each of the components as well as of other adjoining components. The architecture introduced a novelty approach to reduce overhead while extracting longer action proposals, compared to the initial window. In order to mimic human perception, the proposed architecture could act as a base element for a number of higher-level operations, ranging from an auditory assistance devices for visually impaired people, securing vulnerable locations with localized threat predictions, crowd control, fraud monitoring in financial, entertainment, grocery shopping sectors and many more. In the future we will aim to implement the complete architecture and train it on a larger dataset to fully evaluate the synergistic aspects of the proposed components.

CHAPTER 6

References

[1] N. Aafaq, A. Mian, W. Liu, S. Z. Gilani, and M. Shah, “Video Description: A Survey of Methods, Datasets and Evaluation Metrics”, in arXiv:1806.00186v3, 2019.

[2] A. Bhoi, “Spatio-temporal Action Recognition: A Survey“, in arXiv:1901.09403v1, 2019.

[3] Z. Shou, D. Wang, and S.-F. Chang, “Temporal Action Localization in Untrimmed Videos via Multi-stage CNNs”, in arXiv:1601.02129v2, 2016.

[4] S. Buch, V. Escorcia, C. Shen, B.Ghanem and J. C. Niebles, “SST: Single-Stream Temporal Action Proposals”, in CVPR, 2017.

[5] V. Escorcia, F. C. Heilbron, J. C. Niebles and B. Ghanem, “DAPs: Deep Action Proposals for Action Understanding”, in ECCV, 2016.

[6] D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri, “Learning Spatiotemporal Features with 3D Convolutional Networks”, in arXiv:1412.0767v4, 2015.

[7] R. C. Staudemeyer and E. R. Morris, “Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks”, in arXiv:1909.09586v1, 2019.

[8] S. Ji, W. Xu, M. Yang, and K. Yu, “3D Convolutional Neural Networks for Human Action Recognition”, in IEEE, 2013.

[9] H. Rezatofighi, N. Tsoi, J. Y. Gwak, A. Sadeghian, I. Reid and S. Savarese, “Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression”, in arXiv:1902.09630v2 2019.

[10] J. Hosang, R. Benenson and B. Schiele, “Learning non-maximum suppression”, in arXiv:1705.02950v2 2017.

[11] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, “Sequence to sequence-video to text”, in IEEE ICCV, 2015.

[12] Z. Tianyu, M. Zhenjiang and Z. Jianhu, “Combining CNN with Hand-Crafted Features for Image Classification”, in IEEE, 2018.

[13] L. Nanni, S. Ghidoni and S. Brahnam, “handcrafted vs. non-handcrafted features for computer vision classification”, in Elsevier, 2017.

[14] Stanford Vision Lab, Stanford University and Princeton University, (2019, October 13), Image Net, http://www.image-net.org.

[15] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko and T. Darrell, “Long-term Recurrent Convolutional Networks for Visual Recognition and Description”, in arXiv:1411.4389v4, 2016.

[16] R. Arandjelovic and A. Zisserman, “All About Vlad”, in IEEE, 2013.

[17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision”, in arXiv:1512.00567v3, 2015.

[18] A. Rohrbach, M. Rohrbach, N. Tandon and B. Schiele, “A Dataset for Movie Description”, in CVPR, 2015.

[19] K. Soomro, A. R. Zamir and M. Shah, “UCF101: A Dataset of 101 Human Actions Classes from Videos in The Wild”, in arXiv:1212.0402, 2012.