8-1-2019

Human Activity Recognition and Prediction using RGBD Data

Paul Dixon Coen
Southern Illinois University Carbondale, pcoen142@gmail.com

HUMAN ACTIVITY RECOGNITION AND PREDICTION USING RGBD DATA

by

Paul Coen

B.S., Southern Illinois University, 2018

A Thesis
Submitted in Partial Fulfillment of the Requirements for the
Master of Science Degree

Department of Computer Science
in the Graduate School
Southern Illinois University Carbondale
August 2019

THESIS APPROVAL

HUMAN ACTIVITY RECOGNITION AND PREDICTION USING RGBD DATA

by

Paul Coen

A Thesis Submitted in Partial

Fulfillment of the Requirements

for the Degree of

Master of Science

in the field of Computer Science

Approved by:

Dr. Banafsheh Rekabdar, Chair

Dr. Henry Hexmoor

Dr. Tessema Mengistu

Graduate School
Southern Illinois University Carbondale
June 24, 2019

AN ABSTRACT OF THE THESIS OF

Paul Coen, for the Master of Science degree in Computer Science, presented on June 24, 2019, at Southern Illinois University Carbondale.

TITLE: HUMAN ACTIVITY RECOGNITION AND PREDICTION USING RGBD DATA

MAJOR PROFESSOR: Dr. Banafsheh Rekabdar

Being able to predict and recognize human activities is an essential element for us to effectively communicate with other humans during our day to day activities. A system that is able to do this has a number of appealing applications, from assistive robotics to health care and preventative medicine. Previous work in supervised video-based human activity prediction and detection fails to capture the richness of spatiotemporal data that these activities generate. Convolutional Long short-term memory (Convolutional LSTM) networks are a useful tool in analyzing this type of data, showing good results in many other areas. This thesis' focus is on utilizing RGB-D Data to improve human activity prediction and recognition. A modified Convolutional LSTM network is introduced to do so. Experiments are performed on the network and are compared to other models in-use as well as the current state-of-the-art system. We show that our proposed model for human activity prediction and recognition outperforms the current state-of-the-art models in the CAD-120 dataset without giving bounding frames or ground-truths about objects.

ACKNOWLEDGMENTS

I would like to acknowledge all of the advisors, professors, friends, and family who have helped me to be where I am today. Without all of your help, I could never have made it this far. Thank you.

DEDICATION

For my mother Janet Coen, in memoriam. Per aspera ad astra.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Human action detection and prediction is an important task fit for modern machine learning approaches. These tasks are defined as the ability for a system to either recognize actions of humans in a given environment or the ability of a system to predict those actions a given set of time in the future. In recent years, many approaches have helped to improve this field greatly, however further improvements can be made. In particular, recent improvements to network layers have helped to bolster current accuracy measures even higher.

Prior methods focus has been in a two-front effort. The first was focused around image generation and predicting these images over time based on the current and prior images the system has seen. The second method is a form of label prediction in which a system is given sets of labeled images and is asked to either give labels for unlabeled images or predict future labels. Many methods have been used to do this throughout the past 7 years. Approaches that have been attempted in this time include things such as Conditional Random Fields (CRFs) [1] [2] [3], Salient Proto-Objects [4], Support Vector Machines (SVMs) [5] [6], Hidden Markov Models (HMMs) [7] [8], Stochastic Grammar [9], Convolutional Neural Networks (CNNs) [10] [11] [12] [13] [14] [15], Recurrent Neural Networks (RNNs) [16], Glimpse Clouds [17], networks based on Convolutional Long Short-Term Memory layers (Convolutional LSTMs) [18] [19], and a few other hybrid neural network architectures described in [20] [21]. Some issues with these systems are that they either require large structures to represent the given data, or they require large network structures to interpret the data.

Convolutional LSTMs were originally developed in [22] as a system to be used for weather forecasting. Some recent works in human action prediction and detection have used

Convolutional LSTMs as a way to increase accuracy of both methods mentioned above. The efforts in this thesis are focused on the second method, label detection and prediction, using a modified Convolutional LSTM network. A focus is made on the CAD-120 dataset due to its ease of use and possible accuracy improvements. The data shows that this network is both good at predicting and detecting tasks, rivaling the state-of-the-art model in some categories. Another major advantage is that this system utilizes a small number of layers allowing it to be applied to a larger variety of devices.

CHAPTER 2

PRIOR WORK

Human activity detection and prediction is a field rich in many types of data. Others have explored this variety of data in many meaningful ways that have aided in future developments. Many models to detect and predict different types of activities have been developed for RGBD images, skeleton data, object information, and many other inputs. The models developed using these inputs include such things as various types of Conditional Random Fields (CRFs) [1] [2] [3], Salient Proto-Objects [4], Support Vector Machines (SVMs) [5] [6], Hidden Markov Models (HMMs) [7] [8], Stochastic Grammar [9], Convolutional Neural Networks (CNNs) [10] [11] [12] [13] [14] [15], Recurrent Neural Networks (RNNs) [16], Glimpse Clouds [17], networks based on Convolutional Long Short-Term Memory layers (Convolutional LSTMs) [18] [19], and a few other hybrid neural network architectures [20] [21]. Each model poses its own strengths and weaknesses when compared with one another. More details on these methods are given below in each section.

2.1 Graph-based Approaches

Many prior methods of human activity detection and prediction began with looking at graph-based models. One early technique, used by the authors of [7], was to use a hierarchical maximum entropy Markov model. In this system, the authors were able to infer the skeleton data from RGBD photos to find a given person’s body pose and motion information. Using this information, they were able to develop and train a hierarchical graph that can detect what a person is doing based on 12 unique activities. Another venture into the use of Markov models and other systems was done utilizing a modified version of this approach. In [5], the authors focus was to utilize a combination of RGBD photos alongside skeleton data, object information, and interactions between these two to predict and detect given activities going on in a system. To do this, they used a Markov Random Field to model their system as a graph with interactions. To determine classification for a dataset, they were able to take the output of this model and classify it with both an SVM and a k-Nearest Neighbor classifier. The final score for their classification was determined by the sum of these two results. Prediction was done using a similar method. They set some of the initial benchmark results for the CAD-120 dataset used and developed in [5]. Following this, the authors of [6] tried to strip away any unneeded parts of the model in [5]. They ended up focusing solely on SVMs and unstructured graphs. This simplified system achieved higher accuracies than the original by a margin of around 3%.

In [1], the authors tried a different graph-based approach focusing around CRFs. These are a type of graph by which all activities can be viewed as a sequence of trees. These trees describe what is going on at a given timestep t. Predicting future frames becomes of task of building new sets of trees that are likely to arise from what has been seen in prior trees. The use of CRFs has around the same results as prior systems when it comes to accuracy. More specialized CRFs are developed in both [2] and [3]. With [2] the focus was to model and predict a person's trajectories of their pose to infer future actions. In [3], they used a Gaussian Process to reduce the amount of data a human is represented with and predicted their actions based on this reduction.

The authors of [8] revisited older work on Markov models in [5] & [7] and revealed that the best results with this system had not yet been reached. Their work on HMMs was only interested in using skeleton data to detect human activities. Overall, they were able to show that, using this model, both sub-activity and high-level activities can be detected better by a margin of around another 3%, finally pushing overall accuracy above 90%.

New unique methods were developed in both [4] and [9] to attempt to achieve even better results. In [4], the authors developed a system based around Salient Proto-Objects. These are essentially probability maps of what is going to happen in a given photo. Using these probability maps, they were able to apply other techniques, such as the use of SVMs, to classify the images seen into their respective categories. This method was able to be performed without information on ground-truths of the system. It achieved worse accuracy due to this, however it still achieved an impressive 78.2% accuracy on CAD-120. In [9], the authors develop a spatial-temporal And-Or graph represented by stochastic grammar. This grammar describes given subactivities that are made from human actions, objects, and object affordances. Similar to other methods, this gives a hierarchical model to view each event that is occurring. They are able to categorize activities based on a similar structure to decision trees. This tree is then able to predict and detect given activities based on the frame it is currently viewing. The results from this are similar to results from [8].

2.2 Neural Network-based Approaches

A more current approach to human action prediction and detection is to use various types of neural networks. Early work using neural networks began in 2012 with Convolutional Neural Networks (CNNs). These networks are generally good at capturing spatial information; however, they occasionally have trouble seeing temporal interactions. The network described in [12] is one of the first to use three-dimensional convolutional neural networks (3D CNNs) to counteract that issue. Their design is able to use a set of 4 stacked convolutional layers into a fully connected layer. This model's main difference from a normal CNN design is that it was using 3D Convolutions. A few years later, a model developed in [13] utilized a combination of 3D Convolutional layers, however their structure differs due to segmenting their input into multiple segments. These inputs are then fed through separate convolutional layers that all lead and merge to a set of fully connected layers.

In [11], the authors used CNNs for human activity detection. Their main contribution was the ability to apply this to a first-person perspective. Prior methods only used a third person view and ignored the difference in perspective when applying models to a robot performing a similar task. Compared against classifiers that were utilizing k-Nearest Neighbor and Random Decision Forests, CNNs outperformed both methods.

Utilizing similar ideas from [13], the authors of [14] discovered that using the idea of merging and fusing different types of subnetworks together can lead to better learning from a system. In their model, they separate the network into two sets. A temporal set of layers is used to infer data from a perspective over time utilizing 3D convolutional layers. A spatial set of layers was also used to infer data from a spatial perspective using 2D convolutional layers. They fused these two using 3D convolutions as well as 3D pooling to then produce results for human activity detection. At a similar time, [15] utilized a modified auto-encoder model with 3D convolutional layers. This type of model uses a down-sampling technique for the encoder to reduce what information describes the input image. It is then up sampled in the decoder segments of the network to produce an image of the same size where pixels of the new image represent classifications of the pixels. The classification for human activity detection is then derived from a linear SVM applied to the output.

Up to now, all methods mentioned in section 2.2 utilized either RGB or RGBD images coupled with skeleton and/or object information. A unique method from [10] instead used just transformed skeleton information ground truths as input. They first applied various transformations of their system to produce an “image” of the skeleton at a given segment in time.

This consisted of a 3-channel image where the channels represent a different joint of reference and the width and height represent the coordinates and sequence number respectively. This data was then fed to a convolutional layered model to recognize given actions.

On the other side of the spectrum are RNNs. These networks consisted of a series of cells that are fed in as chains into one another. They are good at producing results for temporal interactions, however, usually lack awareness for spatial information. In [16], the authors recognized this weakness and created a structure to attempt to overcome this issue. They utilized various graph structures that can represent models of people and objects through time. These graphs were fed into one another as time passes. An RNN was applied on top of this graph, utilizing it as input. From this, it was able to both predict future action labels and detect current action labels of the system.

A combination of RNN architectures and CNN architectures is used in a hybrid network developed in [20]. They use a set of convolutional input layers that feed into a set of LSTM cells that then feed into a fully connected layer. This allows for the ability to capture important temporal information from the spatial information inferred by the CNN layers going into the RNN layers. As is used by the authors of [17], Glimpse Clouds are a good way to observe a system in which you have partial information of labels. They utilize a similar hybrid approach to the prior methods in [20]. Rather than using LSTMs though, they use a more simplified Gated Recurrent Unit (GRU). Another major difference is that a Glimpse Cloud is composed of a set of predictions, not just one. Each set is composed of a weighted group of outputs from each set. This enables each set to have different predictions based on what they weight priority to over time. Their inputs are also unique in that is does not need fully labeled data to train on. Given partially labeled data, a Glimpse Cloud is able to infer missing labels. They are able to also use not just RGBD images, but also skeleton data to improve results.

[22] is gaining a lot of attention in many applications of prediction and detection. Originally used for precipitation nowcasting, [22] details a new layer for neural networks to utilize to represent spatio-temporal data in a meaningful way. Convolutional LSTMs are introduced here as a way to capture the spatial data from convolutional layers, and the temporal information found by LSTMs. Later papers build on this in many other fields as its layered design is good for things beyond simply forecasting the weather.

One of the first uses of Convolutional LSTM layers outside its original use was in human prediction using unsupervised learning in [19]. The network design described here utilizes a set of cells composed of convolutional layers that use Convolutional LSTM layers as a sort of memory unit for representing events it has seen in the past. These cells are then stacked onto one another where the convolutional layers feed into the next set of convolutional layers and the Convolutional LSTM layers feed back into prior Convolutional LSTM layers. This ensures that prior events are remembered and propagated to all cells. A prediction section of each cell is also generated from the Convolutional LSTM units. In this paper however, they are not looking at predicting labels. They are generating possible future frames. Another more direct approach to predicting future frames using these layers and unsupervised learning was done in [18]. Their design was focused around using the layers directly as the backbone of their network. The authors used a convolutional layer that fed into a set of 7 Convolutional LSTM layers that then fed back into a convolutional layer for image generation. The main focus here was to predict motion in future images. The error of future frames ended up being represented as a blur in images that are generated. So, the further in the future images were generated for, the blurrier they got, thus showing the error seen over time.

The most recent use of Convolutional LSTM layers seen in human action prediction and recognition is in [21]. Here, they use a graph model to represent interactions between object and people in the environment seen. Objects and people are represented as nodes on this graph. The main use of neural networks here is the transitions between timesteps in the graph. The structure is translated and taken as neural network input. Is it then passed through Convolutional LSTM layers which generate the new graph structure one timestep in the future. Tested on CAD-120, it is able to show good results for detection and prediction.

CHAPTER 3

PROPOSED APPROACH

3.1 Network Inputs & Augmentation

Utilizing the Convolutional LSTM layers described in Chapter 2.2 we are able to have the first layer of the Convolutional LSTM as initial network input. Two models were developed to test detection and prediction, one for each. The inputs for the models are composed of a 3-dimensional matrix of photo layers in which the layers are stacked as is seen in Figure 1. The rows and columns are made by the video segment length and the batch size values for these hyperparameters are given in Table *3*. Batches are fed into the system during training and testing. Video segments are derived from the overall videos coming from the dataset. More on the dataset, CAD-120, is described in section 4.1. A skip frame amount is given to reduce unneeded frames in each video and to reduce the overall amount of data that is needed to train the models. From this, each video is divided up into a given set of equal segments of 65 frames each. These video segments are then each taken as one element in the overall batch size. If a video is too short for a segment, the end is buffered with blank frames with padding labels and video frames of zeros. The segment length is kept small to minimize the effect these blank frames cause. If the batch size is not filled on the last segment, it is taken as its own individual batch and not dropped.

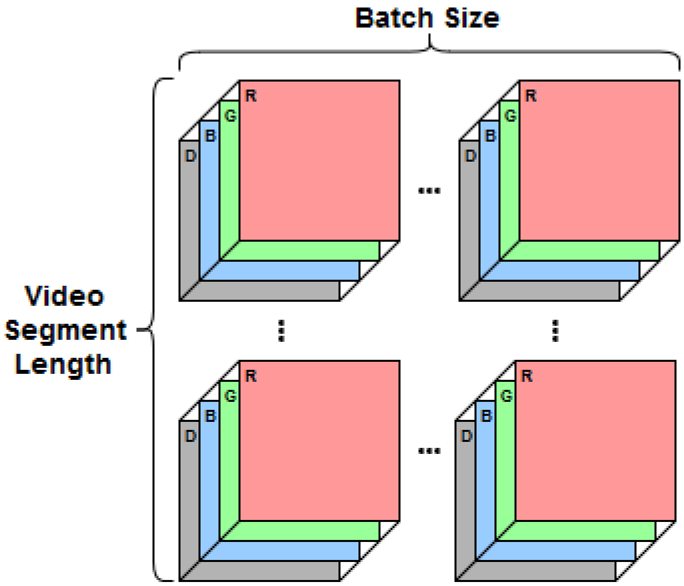

Figure 1: Input structure of network

Augmentation is performed on each image before training and testing to ensure that they are all of equal size. In this case images start as 640 pixels in width by 480 pixels in height, they are converted to 85 pixels in width by 64 pixels in height to maintain the aspect ratio of the photo and reduce data input further. Something to note is before padding in the CAD-120 dataset that the original depth images are not quite the same size as the RGB images. They were padded in the dataset to a maximum value which represents the maximum sensor range of the given depth measurement. These artifacts can be seen clearer in Figure 5. Image normalization was also performed on the depth and RGB images. Both sets of images consist of pixels in a width x height grid with values ranging from 0 to 255 at each point. This causes issues in Convolutional Neural networks occasionally by causing gradient calculations to rise rapidly. The values 0 to 255 were thus mapped to a continuous range from 0 to 1 to minimize how much gradients were influenced.

Labels are also generated from the input files that accompany each class in the dataset for prediction. They are offset 90 frames backwards so that the current frame is described by the label for the future 90th frame. This means that predicted labels are approximately 3 seconds in the future due to the videos consisting of 30 frames per second. For detection, each set of videos is already organized via a folder structure. Each class folder contains sets of videos only for that class for a given individual subject. Each subject's folder also organizes this information further to help divide the given data among them. In the end, the data fed into the system as input and is composed of a 5-dimensional matrix when batches are included. The dimensions consist of batch size, time set value, video layer width, video layer height, and video filter in that order.

3.2 Network Structure

From the given inputs, the data is fed into a series of 4 shrinking convolutional LSTM layers. Figure 2 details how a convolutional LSTM works. Input layers are denoted with I_t to I_{t+1} and so on as more images are given. H and C denote hidden state representations and cell output layers respectively. Other important operators used in the gating functions at time t are defined as i_t (the input gate), f_t (the forget gate), and o_t (the output gate) in Figure 3. Operator $*$ represents a convolutional operation and $\circ$ represents the Hadamard product [22]. The given equations in Figure 3 are representative of the equations needed to represent and update a convolutional LSTM layer.

Following these layers, a common design for convolutional layers is used in which the layers shrink in width and height but increase in depth. These then feed into a fully connected dense layer that is then interpreted as the output labels and batch size. For the predictive model, this was modified to represent a 3-dimensional dense layer with dimensions batch size, output label, a one hot vector representing the current output prediction per frame. A segmented diagram of the model structure can be seen below in Figure 4.

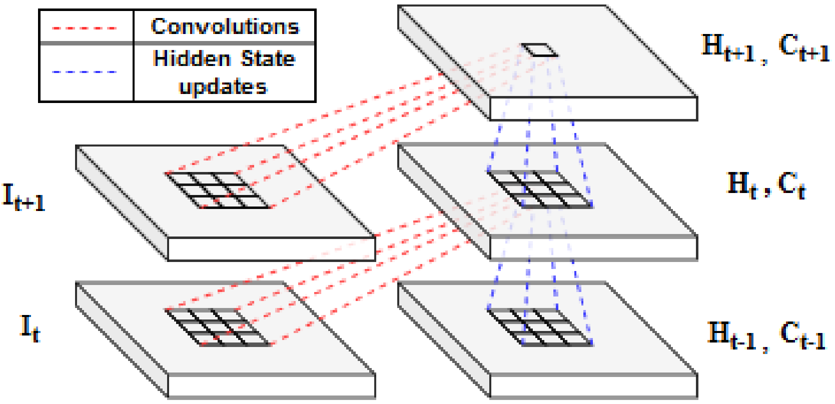

Figure 2: Convolutional LSTM layered structure. Based on Figure 2 in [22].

$$i_t = \sigma(W_{xi} * I_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i)$$

$$f_t = \sigma(W_{xf} * I_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f)$$

$$C_t = f_t \text{ o } C_{t-1} + i_t \circ \tanh(W_{xc} * I_t + W_{hc} * H_{t-1} + b_c)$$

$$o_t = \sigma(W_{xo} * I_t + W_{ho} * H_{t-1} + W_{co} \circ C_{t-1} + b_o)$$

$$H_t = o_t \circ \tanh(C_t)$$

Figure 3: Convolutional LSTM mathematical update operations. Based on equation set 3 in [22].

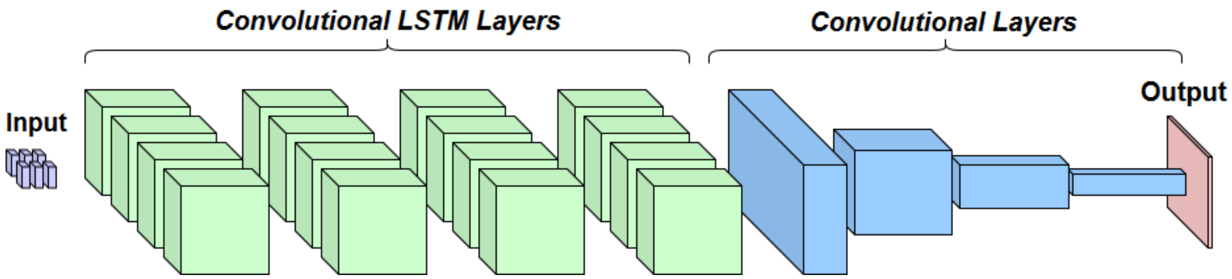

Figure 4: Network Structure Diagram

3.3 Network Outputs & Interpretation

The network output is taken as a set of logit vectors by batch size for detection. These represent the likelihood that the network is giving to current label. Each sample in the batch is described by these vectors. The a softmax function is applied to these logits and the output is then interpreted as the maximum value of the softmax vector to give us the detection/prediction class. We are predicting either the detected high-level activity for the detection issue and the predicted low-level subactivity for the prediction issue. These methods are based on similar methods to the original papers using CAD-120, however we include samples from all subject. A video from each subject-class pair is excluded for testing time to ensure that network is tested on video it has never seen. This ensures that we can see how the models perform new unseen data. A total of 40 of these videos are reserved for testing/validation with 80 remaining to be used for training the models. Implementation of the prediction and detection models was done utilizing the deep learning library TensorFlow in Python3.

CHAPTER 4

EXPERIMENTAL RESULTS & DISCUSSION

4.1 Hardware Specifications

The experiments given in this chapter were trained on Southern Illinois University's local supercomputer Big Dawg's with specifications given in Table 1 as well as my own home computer with specifications given in Table 2. The main node used on Big Dawg was the GPU node. Only half of the resources were used due to the shared nature of the supercomputer.

Table 1: Big Dawg GPU Node Specifications

CPU	Intel Xeon E5-2650 v3
	10 cores at 2 threads each
	2.3GHz base, 3.0GHz turbo
GPU	2 x NVIDIA Tesla K40m GPUs
	12GB of GDDR5 each
	1.43 Tflops of computing power with double precision floating point numbers
RAM	64GB

Table 2: Home Computer Specifications

CPU	Intel i9-9900K
	8 cores at 2 threads each
	3.6GHz base, 5.0GHz turbo
GPU	1 x NVIDIA RTX 2080ti GPU
	11GB of GDDR6
	13.45 Tflops of computing power with double precision floating point numbers
RAM	64GB at 3200MHz

4.2 CAD-120 Dataset

CAD-120 is a publicly available dataset provided by Cornell University's Robot Learning Lab [5]. This dataset was used for experimental evaluation of the prediction and detection abilities of the Convolutional LSTM network model developed in this paper. Pictured in Figure 5 are two samples of image sequences within this dataset. The two sets of sequences seen are composed of RPG and Depth images of the data. RGB images are seen at the top of each set with their corresponding depth image below. These images segments can range in length within this set, from as small as 159 frames, to as large as 1298 frames in the largest sample. This length is annotated in Figure 5 as 1 to n for each video. In both cases, the videos continue past 320 frames, however the major actions that have taken place have already occurred.

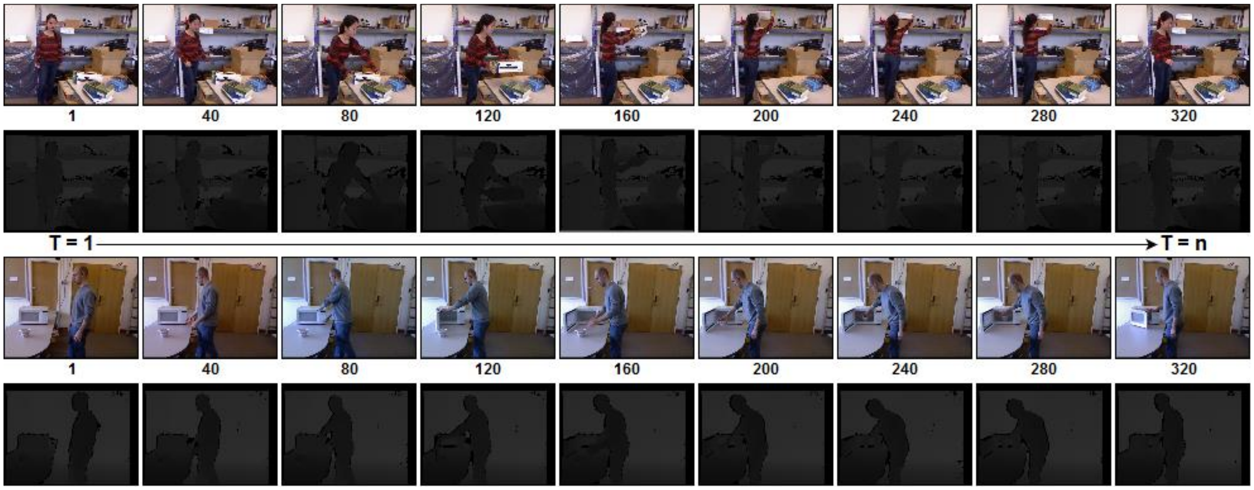

Figure 5: Sample images of two different subject and actions in the CAD-120 dataset. Original paper on this dataset is [5].

Labeling of this dataset comes in two ways. All subjects are their own folder containing all videos of them. The first label, representing high-level activities, comes from the folder structure itself. Every given high-level activity a person performs comes from a given folder name that contains the video sequences. These high-level activities are treated as the label of a video segment when modeling high-level activity detection. The second label we are concerned with, representing subactivities, comes from a file within each high-level set. This file was parsed to label each frame with their activity. These subactivity labels were then offset by 90 frames backwards to have a given frame at time t_k contain the label of frame at time t_{k+90}. This allows for future frames to be predicted by current frames in the model. In particular, 90 frames were used to offset the model to 3 seconds in the future for prediction tasks. This is comparable to other papers that report results for the CAD-120 dataset.

4.3 Results & Parameters

The hyperparameters that were found to produce the best results for both models are reported in Table 3. A major difference between the two models were primarily the maximum frames seen by each one, the batch size, and the learning rate. By reducing the number of frames seen by the prediction model, we were able to reduce the amount of padding frames that needed processes. This was not a major issue for the detection model due to the lack of need for padding labels. A larger batch size was used due to this reduction in data to increase training speed. A lower learning rate also helped to increase accuracy in the prediction model beyond just random chance.

Table 3: Model Hyperparameters

Parameter	Detection Model	Prediction Model
Photo Height Augmentation Size	64	64
Photo Width Augmentation Size	85	85
Maximum Frames	65	25
Batch Size	4	6
Kernel Shape	5 x 5	5 x 5
Learning Rate	0.0001	0.00001

The results in Table 4 show that the model for detection may not be the best model overall; however, it also shows that higher accuracies can be achieved through just the use of RGBD images. Each set in Table 4 show results of models utilizing different amounts of data. The first section shows models using skeleton input, object bounding boxes, ground-truth image segmentation, and RGBD images. Set 2 utilizes the mentioned data in set 1, however it does not utilize ground-truth image segmentation. Set 3 only utilizes either just RGBD images or just skeleton data. This set is the most restrictive when it comes to the possible accuracy that can be achieved. When only considering RGBD data and ignoring information such as skeleton data, frame segmentation, and object bounding boxes, the detection model presented here is the best-in-class with an accuracy of 82.5% and with precision and recall scores of 84.17% and 82.5% respectively. These scores improve upon other models in its set by 1.3% and improves upon the original papers set 2 scores by 1.9%. With respect to other similar set 3 approaches, our model is better due to these increases in accuracy, precision, and recall without an increase in data intake. Other models are able to beat this model though at the expense of more data. Each set in Table 4 is utilizing different amounts of data and is thus not directly comparable to one another with regard to accuracy, precision, and recall.

Table 4: Accuracy of Human Activity Detection

Model Source	Accuracy	Precision	Recall
Set 1	Ground-truth segmentation, object information, and skeleton data		
[1]	**93.5**	**95.0**	**93.3**
[5]	84.7	85.3	84.2
[8]	94.4	-----	-----
Set 2	Without ground-truth segmentation		
[1]	**83.1**	**87.0**	**82.7**
[5]	80.6	81.8	80.0
Set 3	Without ground-truth segmentation & without object bounding boxes		
[4]	78.2	-----	----
[9]	-----	77.0	75.2
[13]	81.2	-----	-----
Convolutional LSTM	**82.5**	**84.17**	**82.5**
Random Chance [5]	10.0	10.0	10.0

The normalized confusion matrix in Figure 6 if given to show that classes are being predicted correctly and which classes are causing issues for the model. Labels given in blue down the diagonal represent positive identifications for each class. Orange labels outside of this represent misclassified classes and what they were classified as instead. All numbers given in the main matrix are normalized from 0 to 100. Values given for 0 are left blank to aid in readability. The row labels represent the correct labeling for each class, and the column labels represent what the network was able to predict. The far right side contains blue columns that represent the per-class precision. The far bottom blue row represents the recall of each class that was seen. Orange percent values in either of these are the incorrect values that were seen and taken away on a per-class basis.

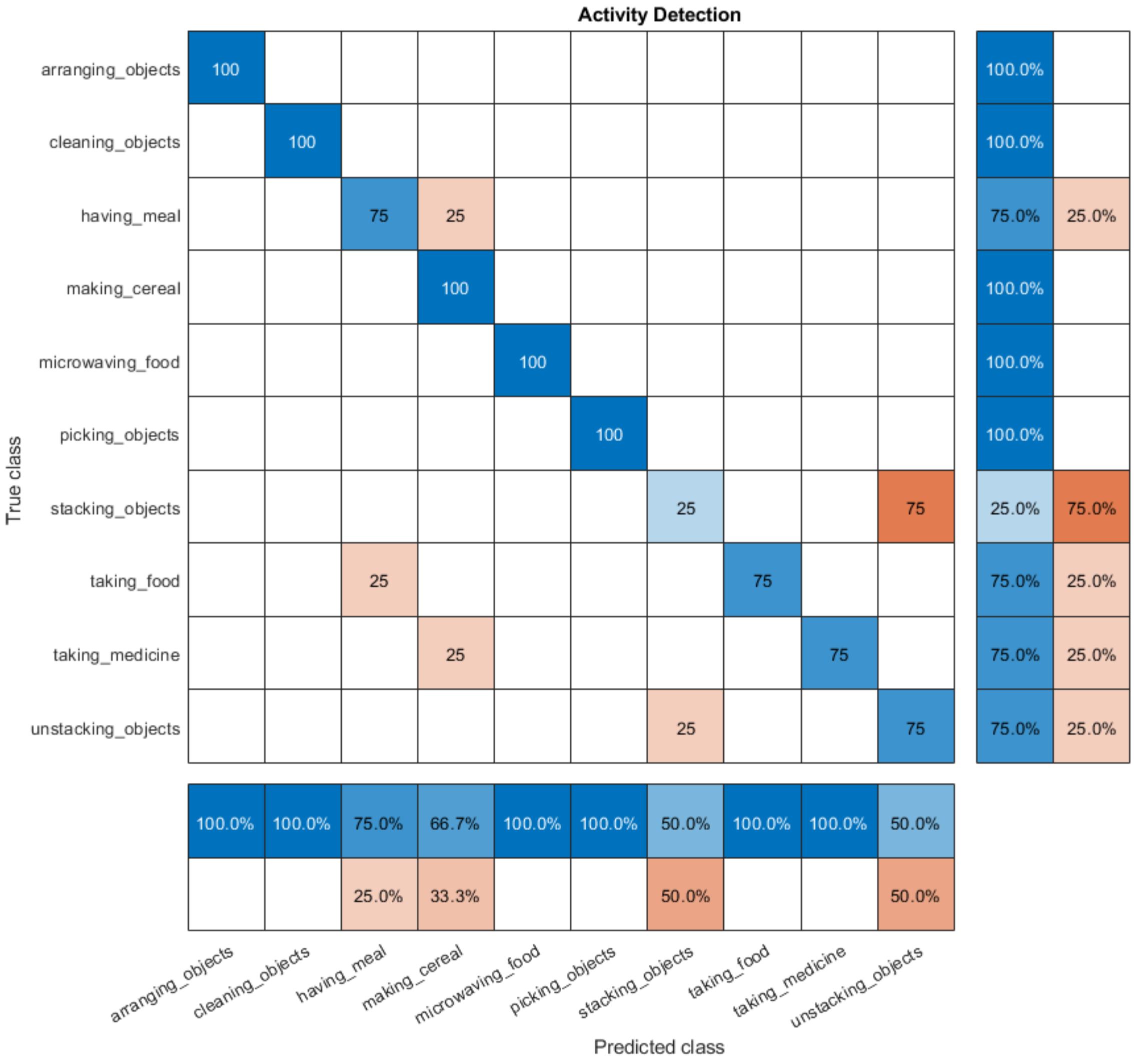

Figure 6: Normalized Confusion Matrix for High-level Activity Detection.

Although high-level labeling faired better than all other comparable models, sublabel predictions did not do nearly as well. As seen in Table 5, the prediction model developed here for 3 seconds in the future was only able to achieve a 41.5% accuracy with even lower precision and recall scores. Although it is utilizing less data and it better than random chance by a large margin, this is still far from achieving good accuracy. Other models are able to do better with an accuracy of 52.1%; however, these low accuracies demonstrate the need for better models and further research in this area.

Table 5: Accuracy of Human Sub-activity Prediction

	Accuracy	Precision	Recall
[1]	49.6	40.6	74.4
[2]	47.7	37.9	69.2
[3]	**52.1**	43.2	**76.1**
[9]	-----	**56.5**	56.6
Convolutional LSTM	41.5	19.43	11.6
Random Chance [5]	10.0	10.0	10.0

CHAPTER 5

CONCLUSION & FUTURE WORK

The need for accurate models of human action detection and prediction is a must with future development of technologies. By being able to predict and/or detect actions of people, robotic systems are able to better respond and adapt to what we are doing and what we are planning to do. Previous work is stated in Chapter 2 detailing the use of the many ways' researchers have tried to model and detect human actions. These methods were attempted with both graph-based models and with neural networks. Chapter 3 then explained the models used in this paper for human activity detection and prediction. Chapter 4 lastly detailed the dataset used and the results observed with the models developed.

Future research in this area should focus around increasing accuracy on predicted subactivities to beyond 50%. Although this may prove difficult, even predicting 3 seconds ahead accurately can be invaluable for countless fields. Another thing that this model could be applied to is a larger, more complete, dataset of human actions with both more actions and subjects. This would help to bolster what the networks are able to predict. Applying this system to an actual robot tasked to aid a human in their current action would also be interesting to work on following this. Other applications of this technology can be applied to medical devices that aid in an assisted living and/or hospital environment so as to aid those given care to patients.

In this thesis, it has been shown that Convolutional LSTM layers can be useful in detecting and predicting human actions with just RGBD data. A penalty in the form of reduced accuracy is taken by just using this data; however, in the case of detection, it is still better than current methods that only utilize RGBD images.

REFERENCES

[1] H. Koppula and A. Saxena, "Learning spatio-temporal structure from rgb-d videos for human activity detection and anticipation," in *International conference on machine learning*, 2013.

[2] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," *IEEE transactions on pattern analysis and machine intelligence,* vol. 38, pp. 14-29, 2015.

[3] Y. Jiang and A. Saxena, "Modeling High-Dimensional Humans for Activity Anticipation using Gaussian Process Latent CRFs.," in *Robotics: Science and systems*, 2014.

[4] L. Rybok, B. Schauerte, Z. Al-Halah and R. Stiefelhagen, "“Important stuff, everywhere!” Activity recognition with salient proto-objects as context," in *IEEE Winter Conference on Applications of Computer Vision*, 2014.

[5] H. S. Koppula, R. Gupta and A. Saxena, "Learning human activities and object affordances from rgb-d videos," *The International Journal of Robotics Research,* vol. 32, pp. 951-970, 2013.

[6] N. Hu, G. Englebienne, Z. Lou and B. Kröse, "Learning latent structure for activity recognition," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[7] J. Sung, C. Ponce, B. Selman and A. Saxena, "Unstructured human activity detection from rgbd images," in *2012 IEEE international conference on robotics and automation*, 2012.

[8] A. Taha, H. H. Zayed, M. E. Khalifa and E.-S. M. El-Horbaty, "Skeleton-based human activity recognition for video surveillance," *International Journal of Scientific & Engineering Research,* vol. 6, pp. 993-1004, 2015.

[9] S. Qi, S. Huang, P. Wei and S.-C. Zhu, "Predicting human activities using stochastic grammar," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[10] C. Li, S. Sun, X. Min, W. Lin, B. Nie and X. Zhang, "End-to-end learning of deep convolutional neural network for 3D human action recognition," in *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 2017.

[11] D. Castro, S. Hickson, V. Bettadapura, E. Thomaz, G. Abowd, H. Christensen and I. Essa, "Predicting daily activities from egocentric images using deep learning," in *proceedings of the 2015 ACM International symposium on Wearable Computers*, 2015.

[12] S. Ji, W. Xu, M. Yang and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence,* vol. 35, pp. 221-231, 2012.

[13] K. Wang, X. Wang, L. Lin, M. Wang and W. Zuo, "3d human activity recognition with reconfigurable convolutional neural networks," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014.

[14] C. Feichtenhofer, A. Pinz and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[15] X. Huang, J. Cheng and X. Ji, "Human contour extraction from RGBD camera for action recognition," in *2016 IEEE International Conference on Information and Automation (ICIA)*, 2016.

[16] A. Jain, A. R. Zamir, S. Savarese and A. Saxena, "Structural-RNN: Deep learning on spatio-temporal graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[17] F. Baradel, C. Wolf, J. Mille and G. W. Taylor, "Glimpse clouds: Human activity recognition from unstructured feature points," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[18] C. Finn, I. Goodfellow and S. Levine, "Unsupervised learning for physical interaction through video prediction," in *Advances in neural information processing systems*, 2016.

[19] W. Lotter, G. Kreiman and D. Cox, "Deep predictive coding networks for video prediction and unsupervised learning," *arXiv preprint arXiv:1605.08104,* 2016.

[20] T. N. Sainath, O. Vinyals, A. Senior and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[21] S. Qi, W. Wang, B. Jia, J. Shen and S.-C. Zhu, "Learning human-object interactions by graph parsing neural networks," in *Proceedings of the European Conference on Computer Vision (ECCV),* 2018.

[22] S. H. I. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong and W.-c. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015.

VITA

Graduate School
Southern Illinois University

Paul D. Coen

pcoen142@gmail.com

Southern Illinois University Carbondale
Bachelor of Science, Computer Science, May 2018

Thesis Paper Title:
Human Activity Recognition and Prediction using RGBD Data

Major Professor: Dr. Banafsheh Rekabdar