Recognizing Actions in Motion Trajectories Using Deep Neural Networks

Kunwar Yashraj Singh, Nicholas Davis, Chih-Pin Hsiao, Mikhail Jacob, Krunal Patel, Brian Magerko

Georgia Institute of Technology
School of Interactive Computing
{kysingh, ndavis35, chsiao9, mikhail.jacob, kpatel311, magerko}@gatech.edu

Abstract

This paper reports on the progress of a co-creative pretend play agent designed to interact with users by recognizing and responding to playful actions in a 2D virtual environment. In particular, we describe the design and evaluation of a classifier that recognizes 2D motion trajectories from the user's actions. The performance of the classifier is evaluated using a publicly available dataset of labeled actions highly relevant to the domain of pretend play. We show that deep convolutional neural networks perform significantly better in recognizing these actions than previously employed methods. We also describe the plan for implementing a virtual play environment using the classifier in which the users and agent can collaboratively construct narratives during improvisational pretend play.

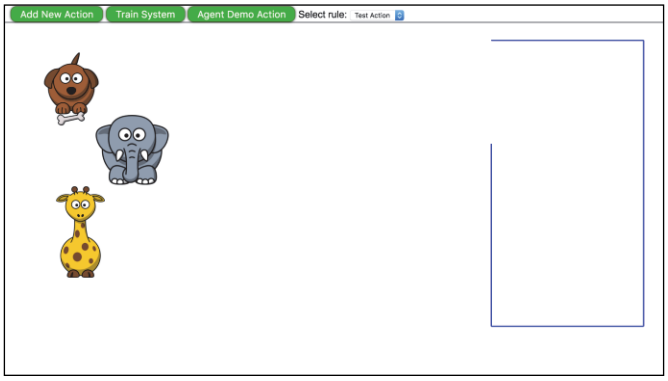


Figure 1: The computational pretend play environment.

Introduction

Pretend play is a universal and foundational aspect of human existence. It serves to strengthen social ties within groups, increase affect between individuals, and allow meaningful learning and practice at creative problem solving (Caillois, 2001; Huizinga, 1950; Power, 1999). Pretend play is therefore a critical part of the human condition within familial and social groups. Understanding play and designing interventions to help facilitate it could have significant impacts on childhood education, therapy, and entertainment. One approach to exploring this problem is developing creative agents designed to engage in pretend play with users. This paper describes our technical progress in developing a co-creative agent that can engage in pretend play with human users.

Pretend play is an improvisational and open-ended creative process, meaning new ideas and activities are dynamically introduced and explored through interaction. Previous work empirically investigating pretend play between dyads (Davis, et al. 2015) found that players gradually co-construct meaning through interaction, involving a tight feedback loop between perception and action in a process the cognitive science literature describes as 'participatory sense-making' (De Jaegher & Di Paolo 2007; Fuchs & De Jaegher 2009). Players recognize stable relationships between their actions and effects in the environment, such as how the other player responds. Through these stable relationships, basic meaning structures emerge referred to as 'nucleus activities' that afford certain types of activities that serve to guide the play interaction moving forward (Davis et al. 2015). As these nucleus activities expand and layers of meaning begin to grow and weave together, a narrative emerges to connect these nucleus activities together.

As a result of the open-ended nature of creative pretend play, there are numerous (potentially infinite) actions and intentions players may utilize during a play session. The huge variety of actions and their associated knowledge requirements make designing an agent for this type of open-

ended creative context a significant challenge. Instead of attempting to encode this type of knowledge into an agent (e.g. using scripts or case-based reasoning), we explore an interactive machine learning solution where users can demonstrate new actions to the system during play. Our approach utilizes data augmentation and deep learning to enable the system to learn how to classify new actions based on a few demonstrations along with user feedback. Utilizing this approach enables a form of crowdsourced knowledge generation where multiple players could add new actions as they become relevant during play sessions, gradually accumulating the knowledge of the creative play agent through its own experience.

Action classification itself is a significant technological challenge, especially when actions need to be understood through direct observation in real-time environments. To narrow the scope of this action classification problem, we implemented the play environment in a simple 2D virtual world, shown in Figure 1, where actions are defined as the movement trajectories of characters within that environment. To test the efficacy of our learning approach, we employ a crowdsourced dataset of 2D actions recently collected by Roemmele et al. that is highly relevant to the domain of play (Roemmele et al. 2016). Our experimental findings suggest that our proposed approach using a deep convolutional neural network is more efficient and accurate for classifying play actions in a 2D environment than the current state of the art.

Background

Pretend play between two or more individuals often involves moving around objects within an environment as well as personifying and attributing intentions to those objects and movements. While narrative is important for play during improvisational play activities, empirical work indicates that the narrative component often emerges through making sense of actions that each player chose to employ at a given time (Davis et al. 2015). That is, narrative serves as a cognitive tool to rationalize and make sense of action sequences coming from multiple parties that are not entirely predictable. Thus, action classification is a critical skill for a co-creative play agent.

The problem of action classification in pretend play is closely related to the decades of work on human activity, action, and gesture recognition from wearable sensors (c.f. Lara & Labrador 2013), mobile phones (c.f. Shoaib et al. 2015), and video footage (c.f. Ziaeefard & Bergevin 2015). Our work embraces the recent trend of applying deep neural networks towards solving this problem (c.f. Cheng et al., 2015). Broadly, deep neural networks have been used in activity recognition with sequential data in recurrent neural networks (RNNs) (Donahue et al. 2015; Venugopalan et al. 2015), and with spatio-temporal volumetric data in convolutional neural networks (CNNs) (Ma et al., 2016; Liu et al. 2016; Wu et al. 2016; Deng et al. 2015). Our approach similarly used spatial data to classify actions from the Charades dataset achieving a higher accuracy than the recurrent neural network approach.

Roemmele et al. (2016) describe early work by Heider & Simmel (1944) that demonstrate how humans attribute intentionality to inanimate objects moving in a simple 2D environment. Heider & Simmel developed a short film that depicted different geometric shapes moving in 2D around a rectangle with a door-like extension. They asked participants in the study, to view the film and describe what they thought was happening. The results overwhelmingly indicated that individuals tended to personify the actions of inanimate objects and weave them into a rich narrative with emotion and social relationships. The interpretation of these movements was shown to be dependent on characteristics of the motion and the current narrative context (eg. distinguishing between fly, attack, turn, etc. based on the specific motion trajectory and narrative context).

Roemmele et al. (2016) developed an updated version of the experiment to collect two crowd-sourced datasets: the Charades dataset, which consisted of animations of triangles performing a single action (defined by a verb either affecting one or two actor triangles), and the Theatrical dataset, which consisted of triangles performing recognizable actions. In their analysis, Roemmele et al. compared two distinct machine learning approaches, a spatiotemporal bag of words model and a recurrent neural network, to classify which action or set of actions corresponded to the trajectories of the actors in both of their datasets. Their models achieved 12.5% using bag-of-words and 25% using recurrent neural networks respectively. In this paper, we compare our approach, convolutional neural networks, to the methods proposed by Roemmele et al. on the Charades dataset to evaluate its effectiveness and utility for action classification in a pretend play environment.

System Design

Figure 1 shows the online pretend play environment in which the creative play agent can interact with users in real-time. This online environment contains a 2D virtual playground with characters that the user can move around. The characters that can be manipulated and moved by users are cartoon animal images inspired by the set of toys used during our empirical investigation of play (Davis et al. 2015). These animal characters were found to encourage a playful mindset and allow for more explicit and intentional attribution to the movements during play.

Creating new actions is critical for improvisational play since there is a wide variety of actions users may want to employ during a play session. Users can add a new action to

the agent's knowledge base by selecting the appropriate button, labeling their desired action, and proceeding to demonstrate its performance. The system was designed to learn the target action with a high degree of accuracy in a minimal amount of demonstrations to reduce the training burden on the user.

Once an action has been demonstrated, the agent can showcase its learned capabilities in the virtual playground with the player. For instance, when a player selects an action for the system to perform, the agent selects a character on the screen and moves it in the path specified by the policy learned for that action. After the agent performs an action to demonstrate its understanding, the user can then confirm or deny whether the demonstration accurately portrayed the intended action by voting up or down with feedback buttons to provide supervision to the learning process.

The online environments' full integration with the deep learning modules described below allows for our deep learning-based model to learn and classify actions beyond those in the Charades dataset used in the evaluations described here. Once the user defines and performs an action, the trajectory of the actor in the playground is sent to the neural network for training. The backend of the pretend play environment consists of three core components related to the neural network architecture: Data Congealing, Motion Trajectory Classification, and Motion Trajectory Generation. An additional input module takes in a bitmap image that contains the motion shape and resizes it to be sent to the Data Congealing module.

The *data congealing module* we employ takes in the input image and generates similar images to increase the amount of training data for the system. *This is particularly useful in the domain of pretend play since it reduces the amount of demonstrations required for teaching the system a new action*. The data congealing modules uses two techniques: 1) applying random rotations, translations, and left to right reflections onto the user input; and 2) using reinforcement learning algorithms to generate trajectories which are similar to the actual input but deviate slightly in shape. For example, if a circular shape is fed into the congealing module, it would output circles of different sizes and oval shapes that are similar to the circle but not the same. Jittering the input data in this manner ensures that the system does not over-fit to the sample data received from the user and offers a greater degree of generalization (Yu et al. 2015).

Once the appropriate training data has been generated, it is sent to the *classification* and *generation modules* respectively. The classification module uses a deep convolutional neural network to classify the motion trajectory and help establish a shared context with the user. The generation module uses a Deep Convolutional Generative Adversarial Network (Goodfellow et al. 2014) to generate motion trajectories from the previously learned inputs to be outputted onto the playground as the co-creative agent's action. These modules work together to understand the meaning behind the motion trajectories of the user's actions and enable the agent to generate its actions during pretend play. To learn new actions, transfer learning is utilized in the above modules where the trained network weights are reused to retrain on the new action-label pair (Azizpour et al. 2015). This allows for incremental learning while still retaining the previously learned knowledge.

The reason behind having a classifier in a co-creative pretend play system is that the system must be able to recognize the user's current action accurately in order to understand their intention and construct a meaningful narrative. In order to engage the user effectively, the system must be able to recognize these trajectories with a high accuracy so that both the parties can move forward after they have successfully established a shared context. Once high accuracy in action recognition has been achieved for individual actions, the next step is to focus machine learning on understanding how actions are performed together and sequenced according to the narrative that is dynamically emerging during pretend play. If the system is unable to understand how these individual actions are performed together, it may lead to an ineffective play partner where the agent would be unable to collaboratively grow the nucleus activity (Davis et al. 2015).

Neural Network Architecture

We sought to apply a deep learning approach to the Charades dataset in order to evaluate its effectiveness and compare its utility to methods proposed by Roemmele et al. for action classification in a pretend play environment. To that end, we changed the framing of the problem from sequence classification to image recognition. We predicted that a deep learning approach with convolutional neural networks would yield better classification accuracies because of these networks' recent successes with images and sketches (Yu, et al. 2015; Simonyan K. & Zisserman A 2014; Szegedy et al., 2015).

The structure and functional organization of convolutional neural networks are inspired from the biology of the human eye (LeCun, Y. & Bengio, Y. 1995). They consist of multiple learnable filters arranged in layers, which each extract relevant features from input images, just as the visual cortex has different layers that each have unique specializations in processing visual information. The cognitive argument for using convolutional neural networks in a co-creative play agent is that using such networks would resemble how classification and recognition would occur in the human vision system. Furthermore, an extensive amount of previous research has addressed action and gesture recognition from camera video data using convolutional neural networks (Tran et al. 2014). However, our work emphasizes the application of deep learning methods for recognizing motion trajectories as opposed to the recent emphasis on image caption generation.

For the purpose of classifying motion trajectories, we borrowed from various other convolutional neural network architectures to assemble a classifier that can work with 2D images without texture information. As shown in Figure 2, we ended up modifying the VGG CNN-S model by removing Local Response Normalization as they perform well with images that have textural information but not well with edges or sketches (Yu et al. 2015; Krizhevsky et al. 2012). We refer to this model as Deep-Play and this model works best with 2D motion trajectories rather than 2D images with texture information. The input is a 224 by 224 image and it outputs the scores for 32 categories of actions. The overall architecture is specified below:

Layers	
	Input (3 x 224 x 224) image
1	Conv (Filters: 96, Filter Size = 7 x 7, stride: 2)
	Max Pool (Pool Size: 3, Stride: 3)
2	Conv (Filters: 256, Filter Size = 5 x 5)
	Max Pool (Pool Size: 2, Stride: 2)
3	Conv (Filters: 512, Filter Size = 3 x 3, Pad: 1)
4	Conv (Filters: 512, Filter Size = 3 x 3, Pad: 1)
5	Conv (Filters: 512, Filter Size = 3 x 3, Pad: 1)
	Max Pool (Pool Size: 3, Stride: 3)
6	FC (neurons: 4096, dropout = 0.5)
7	FC (neurons: 4096, dropout = 0.5)
	Softmax (classes = 32)

Table 2: The convolutional neural network architecture.

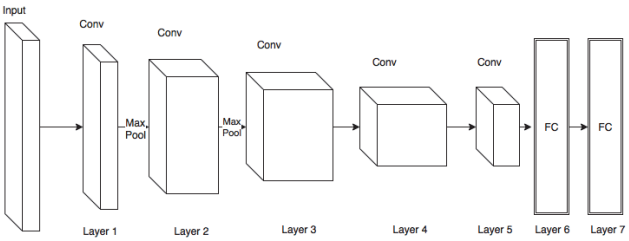


Figure 2: The convolutional neural network after modifying VGG CNN-S

In the following sections we describe the results of our experiments evaluating our proposed approach. Based on our results, we argue that convolutional neural networks are a better candidate for classifying the underlying action from motion trajectories as compared to recurrent neural networks used in the past experiments.

Experiments

The dataset we used to evaluate the accuracy of action classification in our system was a publicly available dataset of labeled actions called the Charades dataset. Roemmele et al. collected this data in a crowd-sourced manner, using a game where users perform actions in a 2D virtual environment using simple shapes as characters (such as a triangle, square, circle, etc.). Other players then viewed these recorded actions and guessed the high-level label for the action (similar to the popular game called Charades). Those actions with a high level of agreement among players were added to the database of labeled actions. Example actions include dance, jump, run, accelerate, spin, fly, roll, and roam. This dataset was particularly informative for our target domain of pretend play as the actions covered a wide variety of verbs that could be used when expressing ideas and playing out scenes during pretend play.

There are both one-character and two-character datasets. The one-character data contains the motion trajectories of actions that were created using only one character in the Charades game, whereas the two-characters data contains a consolidated set of the motion trajectories of actions constructed using two characters in the game. There are 2060 one-character animations and 1158 two character animations in the dataset. The animations are represented as a set of (X, Y) coordinates that describe these motion trajectories.

Roemmele, et al. were able to achieve 12.6% accuracy on the one-character data and 25% on the two-character data set. This dataset is an ideal candidate for exploring the power of convolutional neural networks since they mimic human vision and can recognize visual patterns better than other types of neural network. For a successful game of Charades to happen, the actions must be classified accurately so that the system can progress after establishing a shared meaning and continue with the narrative. This process directly transfers to the pretend play domain since the agent needs to understand the current action and build on the action in order to continue the story in a successful play session. This motivated our focus on action classification accuracy in our experiments.

The convolutional neural networks were run individually on each of the sets (i.e. one-character and two-character data). Since each dataset came with a set of testing data, it helped in defining the baseline to compare our classifier with the other classifiers used in the previous work. In addition to the previous results obtained by Roemmele et al. using a spatio-temporal bag-of-words and recurrent neural network approach, we explored other convolutional neural network architectures to offer a comparison with the Deep-Play classifier used in our system. For example, we also tested Google's Inception network with batch normalization, which won the ImageNet 2014 challenge on classifying images.

The training pipeline to the deep convolutional neural network involves preprocessing images to encode spatial information before sending it to the network. This was done by plotting the motion trajectory and creating an image out of it with different colors assigned to each character's motion trajectory so that the neural network learned to differentiate

between motions of different characters. One key difference between our approach and the previous approaches was that we removed the time dimension from the input sequence and only worked with the spatial data. Some example images along with the actions they represent are shown in Table 3 below.

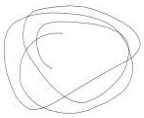
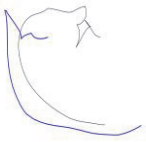
		
Turn	Accelerate	Spin
		
Accompany	Argue with	Mimic

Table 3: The images representing temporal information that is sent to the neural network. Top row shows one-character examples and bottom row shows two-character examples.

During the training phase, we made use of simple data augmentation techniques such as horizontal flipping and random rotating to counter overfitting similar to the working of the congealing module mentioned in the system architecture section. The point to note here is that the previous state-of-the art results obtained on this data set used handcrafted features for constructing the spatio-temporal bag-of-words vocabulary. The features of the spatio-temporal bag-of-words model are described in Table 4.

1-character	Distance, Rotation, Angle, Angle Offset, Velocity, Rotational Velocity, Acceleration, Rotational Acceleration, Jerk, Curvature, Angle Change
2-character	Relative Distance, Relative Angle, Relative Velocity, Relative Acceleration, Relative Jerk, Relative Angle Change

Table 4: The features used to the construct the Bag-of-Words model (Roemmele et al. 2016).

The previous research also made use of recurrent neural networks. These networks can use their internal memory to learn and classify sequences. For this neural network, the input was *distance, rotation,* and *velocity* used to represent the input as a sequence (Roemmele et al. 2016). The two methods attempted in the previous work required some degree of feature engineering and recording extra parameters. In general, these parameters are selected mostly by intuition or through prior research and can restrict the model's accuracy. However, one can bypass recording of these extra parameters if the actions are represented visually. Therefore, our approach using convolutional neural networks aims to provide an end-to-end learning solution without hand-engineering any of the features. This allows for automatically extracting the relevant features from the 2D lines, as their meaning could change over time.

Results

The experiments were performed on the Charades dataset mentioned above. For the one-character dataset, the neural network was run for 100 training iterations with a learning rate of 0.001 using stochastic gradient descent with Nesterov momentum of 0.9 for each of the model architectures namely, Deep-Play and Google Inception network with batch normalization (Szedegy et al. 2015). The results below show the classifier's accuracy on the one-character dataset and it includes the results from the previous research where words + LR is the bag-of-words with logistic regression, words + NB is the bag-of-words with Naive Bayes classifier.

Deep-Play: **99.5%**	Google-Inception: 76.8%	Words + LR: 12.6%	Words + NB: 8.5%
1 Layer - RNN: 8.0%	2 Layer - RNN: 7.3%	Baseline: 5.3%	

Table 5: Classification accuracies for one-character dataset.

These results demonstrate that using a convolutional neural network on the image-based representation of the motion trajectories improved the accuracy drastically as compared to the state of the art accuracy of 12.6% using the Words + LR method. Moreover, we can see that Google's Inception network was only able to achieve an accuracy of 76.8%, despite being the state of the art on the image-net dataset for object recognition in images (Szedegy et al., 2015). This finding supports our hypothesis that detection of motion trajectories requires a network that can work in the absence of texture information. The Deep-Play classifier was able to achieve a 100% recognition rate on the test set except for certain actions such as limp, stumble, creep and drift. Most motion trajectories used to represent these classes where hard to differentiate, as they were highly overlapping. The accuracies are provided below:

			
Limp: 90%	Stumble: 92%	Creep: 90%	Drift: 85%

Table 6: Accuracy on classes that were classified incorrectly

For the two-character dataset, the Deep-Play classifier was trained using ADAM optimizer (Kingma, D. & Ba, J. 2015) and a learning rate of 0.0001. In contrast, stochastic gradient descent gave varying accuracies for each iteration. However, at very high epochs, this optimizer gave results on par with ADAM. The results using the ADAM optimizer are described in the table below:

Deep-Play: **28.70%**	2 Layer - RNN: 25.0%	Words + NB: 22.0%	1 Layer - RNN: 18.5%
Words + LR: 12.5%	Baseline: 5.6%		

Table 7: Classification accuracies for the two-character dataset.

As noted, our classifier performed slightly better than the 2-layer RNN from the previous research. When compared to the high accuracy on the one-character dataset, we can see that Deep-Play and the other classifiers perform poorly on this dataset. Even though with only the spatial representation available, the classifier was still able to improve upon the previous state-of-the-art accuracy of 25% to 28.7%. We provide a more detailed description about the possible causes that lead to a poor performance on this dataset in the discussion section below.

Discussion

The experiments provided us with several insights on the problem of recognizing actions from motion trajectories. The primary insight we found was that the issue of recognizing motion trajectories is more of a computer vision problem than a low-level sequence classification problem as the motion sequences can be represented as an image to account for the spatial information. We also found that convolutional neural networks are a better candidate for classifying such trajectories than recurrent neural networks despite that the time dimension in images was not present.

In the two-character dataset we reported that no classifiers we experimented with gave promising results. This was due to the fact that the examples in the training and test set were highly overlapping as the temporal dimension was not present in the image based representation of these trajectories. For example, actions such as "accompany" and "follow", illustrated in Table 3, were represented using similar motion trajectories in absence of the temporal dimension. Thus, during our experiment the classifier reached varying accuracies for each successive epoch. This problem could be addressed by examining the data to ensure that they do not overlap. The overlap could be due to a bad example or due to the absence of temporal dimension. The temporal information could be encoded using sequence of images rather than a single image used in our experiments. This image sequence can then be classified using spatio-temporal convolutional network. This approach is left as a future work.

Conclusions

This paper described a co-creative pretend play agent designed to interact with users by recognizing and responding to playful actions in a 2D virtual environment. We identified action classification as a primary challenge for a co-creative pretend play agent in order to build shared meaning and co-construct a narrative through interaction. Through our experiments, we found that the actions can be represented as images that capture their spatial relationships and show how convolutional neural networks are effective in recognizing such motion trajectories as compared to recurrent neural networks used in past research.

Acknowledgements

We thank the researchers at ICT for making the Charades dataset publicly available. These experiments would not have been possible without that dataset. This work was supported by NSF grant IIS-1641008.

References

Azizpour, H., Sharif Razavian, A., Sullivan, J., Maki, A., & Carlsson, S. (2015). From generic to specific deep representations for visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 36-45).

Caillois, R. (2001). *Man, play, and games*. University of Illinois Press.

Chatfield, K., Simonyan, K., Vedaldi, A., & Zisserman, A.(2014). Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*.

Cheng, G., Wan, Y., Saudagar, A. N., Namuduri, K., & Buckles, B. P. (2015). Advances in human action recognition: A survey. *arXiv preprint arXiv:1501.05964*.

Davis, N., Comerford, M., Jacob, M., Hsiao, C.-P., & Magerko, B. (2015). An Enactive Characterization of Pretend Play. In *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition* (pp. 275–284).

De Jaegher, H., & Di Paolo, E. (2007). Participatory sense-making. *Phenomenology and the Cognitive Sciences*, 6(4), 485–507.

Deng, Z., Zhai, M., Chen, L., Liu, Y., Muralidharan, S., Roshtkhari, M. J., & Mori, G. (2015). Deep structured models for group activity recognition. *arXiv preprint arXiv:1506.04191*.

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).

Foggia, P., Saggese, A., Strisciuglio, N., & Vento, M. (2014, August). Exploiting the deep learning paradigm for recognizing human actions. In *Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on* (pp. 93-98). IEEE.

Fuchs, T., & De Jaegher, H. (2009). Enactive intersubjectivity: Participatory sense-making and mutual incorporation. *Phenomenology and the Cognitive Sciences*, 8(4), 465–486.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems* (pp. 2672-2680).

Hasan, M., & Roy-Chowdhury, A. K. (2014). Continuous learning of human activity models using deep nets. In *Computer Vision–ECCV 2014* (pp. 705-720). Springer International Publishing.

Heider, F., & Simmel, M. (1944). An experimental study of apparent behavior. *The American Journal of Psychology*, 57(2), 243–259.

Huizinga, J. (1950). *Homo Ludens: A study of the play -- element in culture. a study of the element of play in culture*. Routledge.

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).

Kingma, D., & 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).

Lara, O. D., & Labrador, M. A. (2013). A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys & Tutorials*, 15(3), 1192-1209.

LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995.

Liu, Z., Zhang, C., & Tian, Y. (2016). 3D-based Deep Convolutional Neural Network for action recognition with depth sequences. *Image and Vision Computing*.

Ma, M., Fan, H., & Kitani, K. M. (2016). Going Deeper into First-Person Activity Recognition. *arXiv preprint arXiv:1605.03688*.

Peng, X., Zou, C., Qiao, Y., & Peng, Q. (2014). Action recognition with stacked fisher vectors. In *Computer Vision–ECCV 2014* (pp. 581-595). Springer International Publishing.

Power, T. G. (1999). *Play and exploration in children and animals*. Psychology Press.

Razavian, A., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 806-813).

Roemmele, M., Morgens, S.-M., Gordon, A. S., & Morency, L.-P. (2016). Recognizing Human Actions in the Motion Trajectories of Shapes. In *Proceedings of the 21st International Conference on Intelligent User Interfaces* (pp. 271–281).

Shoaib, M., Bosch, S., Incel, O. D., Scholten, H., & Havinga, P. J. (2015). A survey of online activity recognition using mobile phones. *Sensors*, 15(1), 2059-2085.

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

Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems* (pp. 568-576).

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). Rethinking the Inception Architecture for Computer Vision. *arXiv preprint arXiv:1512.00567*.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2014). Learning spatiotemporal features with 3d convolutional networks. *arXiv preprint arXiv:1412.0767*.

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, K., Wang, X., Lin, L., Wang, M., & Zuo, W. (2014, November). 3D human activity recognition with reconfigurable convolutional neural networks. In *Proceedings of the ACM International Conference on Multimedia* (pp. 97-106). ACM.

Wang, L., Qiao, Y., & Tang, X. (2015). Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4305-4314).

Wu, D., Pigou, L., Kindermans, P. J., Nam, L. E., Shao, L., Dambre, J., & Odobez, J. M. (2016). Deep Dynamic Neural Networks for Multimodal Gesture Segmentation and Recognition. *IEEE Explore*.

Yu, Q., Yang, Y., Song, Y.-Z., Xiang, T., & Hospedales, T. M. (2015). Sketch-a-net that beats humans. In *Proceedings of the British Machine Vision Conference (BMVC)* (pp. 1–7).

Zeng, M., Nguyen, L. T., Yu, B., Mengshoel, O. J., Zhu, J., Wu, P., & Zhang, J. (2014, November). Convolutional neural networks for human activity recognition using mobile sensors. In *Mobile Computing, Applications and Services (MobiCASE)*, 2014 6th International Conference on (pp. 197-205). IEEE.

Zhang, L., Wu, X., & Luo, D. (2015, December). Recognizing Human Activities from Raw Accelerometer Data Using Deep Neural Networks. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)* (pp. 865-870). IEEE.

Ziaeefard, M., & Bergevin, R. (2015). Semantic human activity recognition: a literature review. *Pattern Recognition*, 48(8), 2329-2345.