University of Arkansas – CSCE Department
Capstone [I] –Final Proposal– Fall 2021
Automatic Action Recognition in Videos

Team 13: Garrett Bartlow, Braxton Parker, Daniel Miao,

Joshua Stadtmueller, Jonathan Zamudio

Mentor: Prof. Khoa Luu

Abstract

Automatic action recognition is one of the primary tasks in video understanding. It has various practical applications such as human behavior analysis, virtual reality, and gesture recognition. Advancement in artificial intelligence has resulted in the studying of deep learning techniques. Deep learning is the technique to perform machine learning based on the structure and inspiration of the human brain. In the deep learning era, there are many methods that have been proposed to address the problem of action recognition. The objective of this project is to successfully apply the techniques used in the Temporal Shift Module (TSM) to solve a real-world problem in automatic action recognition. This will be done using technologies like deep learning libraries (PyTorch), Computer Vision and Video Processing (OpenCV) and Scikit-Learn. The goal is that by offering a successful application of the TSM, applications for more advanced problems may be inspired or extrapolated such that there is an obvious and non-trivial benefit in the community.

1.0 Problem

The technology of video understanding has increased dramatically over the past decade. With the development of artificially intelligent systems, there now exists a domain of problems that can be solved. For example, trying to expand the accessibility of using computers for those who traditionally could not, is now possible.

The requirement for usage of a mouse and keyboard to operate a computer is not always ideal as they are not completely intuitive to use and/or not able to be used. Some patients in physical therapy cannot utilize computer programs because of the requisite use of a mouse and keyboard. So, not having an optimized solution for

2.0 Objective

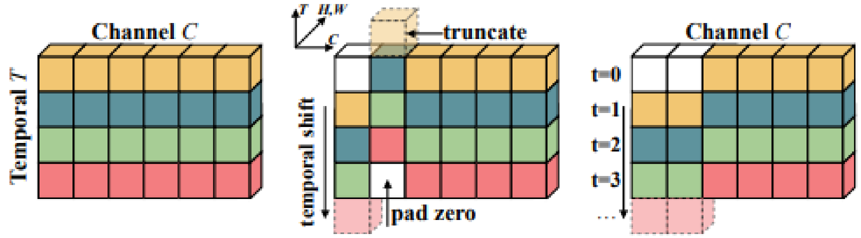
The first main objective is to create a GUI using the model-view-controller method. We want to also utilize deep learning libraries and video processing software to recognize gestures according to our standards. The last main objective designs our final project so that it may act as a launching platform for more complex variations in the future.

Background

2.1 Key Concepts

Temporal Shift Module (TSM) utilizes the temporal dimension to manipulate data in order to achieve 3D CNN (Convolutional Neural Network) results with 2D CNN complexity. A convolution neural network is an artificial neural network used most in video and image recognition [4]. TSM is ultimately a technique using these CNN's where some data is taken in the form of video, then filtered into a collection of images. From these images, the already trained model predicts the current gestures shown in each from, if such a gesture exists from the pre-trained model [1]. There is an extension of the TSM module in which gestures can be classified and identified in real time at the expense of only a cache holding 1/8 of current features used in the model [1].

The application of the online TSM uses uni-directional TSM in which there is only a shift in the feature data from previous frames until current frames (Frames being the current "picture"). This is differentiated from bi-directional TSM which uses futures frames to influence the current frames on the feature data. As seen in figure 2, there is a slight information difference between uni-directional and bi-directional models as the number of spatial dimensions grow.



(a) The original tensor without shift. (b) Offline temporal shift (bi-direction). (c) Online temporal shift (uni-direction).

Figure 2: From [1]: Temporal Shift Module (TSM) performs efficient temporal modeling by moving the feature map along the temporal dimension. It is computationally free on top of a 2D convolution but achieves strong temporal modeling ability. TSM efficiently supports both offline and online video recognition. Bi-directional TSM mingles both past and future frames with the current frame, which is suitable for high-throughput offline video recognition. Uni-directional TSM mingles only the past frame with the current frame, which is suitable for low-latency online video recognition.

Within the implementation of TSM, Python, OpenCV, PyTorch, and Scikit-Learn are used. OpenCV is an open-source library that deals with real-time computer vision. It can be used to process images and videos to identify different actors within frames such as objects, people, handwriting, and even hand gestures. OpenCV supports a wide variety of languages which includes python3.

PyTorch and Scikit-Learn are both machine learning libraries for Python that enable the manipulation, construction, and analysis of data. For the purview of this project, the data gathered from OpenCV is then translated into different array types such that it can then be used with Scikit-Learn. The difference, however, between PyTorch and Scikit-Learn is that PyTorch is much more suitable for deep learning and is used extensively here for the implementation of the TSM model. However, both frameworks are used at some point within the project.

2.2 Related Work

A team at MIT has done research which is related to what we are trying to accomplish here, using TSM for video recognition. The MIT team's paper [1] has a focus on the utilization of TSM to increase efficiency and their team implemented it with Google Maps, where as our focus is finding practical solutions and implementing it with PowerPoint gestures.

There have also been strides to translate [2] sign language using gesture recognition in video processing. Elmahgiubi's team found that a fitted glove with sensors combined with a CNN allowed for the recognition of most of the letters in the ASL alphabet. The ASL vocabulary is incredibly dependent on the subtle movement and location of the fingers. So, high accuracy and precision is necessary for a successful model. The fitted glove and proposed model yielded a 96% average accuracy as well as recognizing 20 out of 26 letters [2]. Another team proposed a similar system in which infrared images are used to feed to the CNN [5]. While the implementation of each CNN may be different, the input parameters are surely different as one team used sensor data, and another used infrared imaging. Tao's team [5] achieved a 99.7% average accuracy when testing all 24 alphabet gestures with 5 different patients. So, while it is important to note the technology of CNNs, the input selection plays a role.

Not only are known gestures configurable in video recognition, but also a [3] team has discovered how to predict unknown gestures using similar techniques as well as a cognitive behavioral model. More related to TSM, Benitez-Garcia's team proposed system inspired by TSM in which specifically TSM is inserted to a temporal shift network (TSN) where TSN operates only on a sequence of clips within a video instead of the entire video. The final video-level prediction is then considered a summation of accumulation of each shorter clip sequence [6].

These other works have significance to the proposed application because they allow for inspiration and proof-of-concept ideas for new projects. More specifically, the development of the sensor fitted glove shows that wearable technology can be integrated with the ideas put forth in the TSM paper [1]. The advancement of one facet of technology can also lead to the development of new technologies as shown by Benitez-Garcia [6].

3.0 Design

3.1 Requirements and/or Use Cases and/or Design Goals

The completed implementation of the application results in the computer responding correctly to a given command. Ex, if “swipe left” is shown on camera, only the command for that gesture will be executed. Any non-identifiable gesture will not result in a command. Only one person doing a gesture will be identified.

The GUI for the application offers clear and concise interaction and readability for the features highlighted above. For demonstration purposes, gesture control will be used to control different actions in PowerPoint. For example, using the swipe gesture will change to a different slide. PowerPoint is not the only software capable of being controlled with gesture commands, but it is the most relevant to the project at hand. The webcam and PowerPoint must be synchronous enough such that a presentation demonstrating gesture control can be achieved. The GUI must not crash given the above conditions.

3.2 High Level Architecture

At the root of the design, the application consists of a GUI which will display the webcam and some interactive medium next to it as shown in Figure 2.

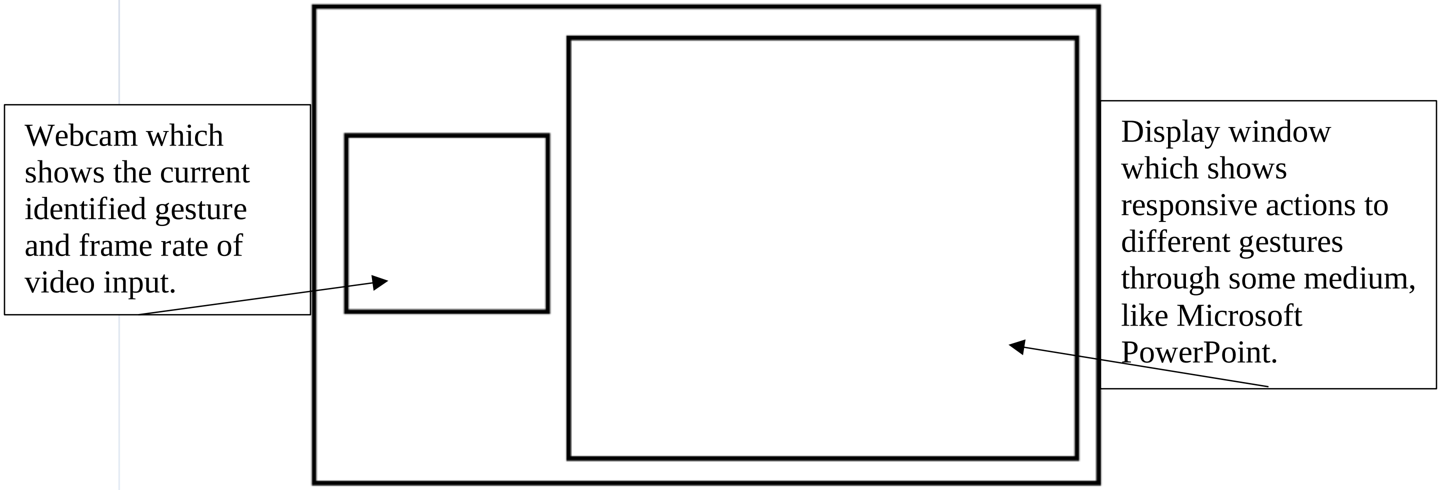

Webcam which shows the current identified gesture and frame rate of video input.
Display window which shows responsive actions to different gestures through some medium, like Microsoft PowerPoint.

Figure 2. Base level design for GUI application

The application using the OpenCV framework will consist of an “embedded” PowerPoint application and a running webcam. Whenever a recognized gesture is given which will be seen in the left-most window, the display window on the right will react according to the designed action designated by the team. Within the webcam window there will be a gesture identification as well as a real-time accurate frame rate counter which shows the input parameters to the TSM framework (The number of frames per second is how much data is inserted into the real time model. It is important to note that this value is not static).

3.3 Risks

Risk	Risk Reduction
Team is unfamiliar with video recognition software	Having the main code be pre-created, where we just focus on the implementation. The base code is a large portion of the implementation. In-depth study of the base code will be conducted such that each member understands fully the technology behind the application.
Unauthorized Manipulation of another company software	Implementing it in a manner to not interfere with the actual software.

3.4 Tasks

1. Understand/gain background about using the hand gestures inputs and controlling outside applications/software.
2. Define exactly what actions/controls will be implemented.
3. Determine design of GUI for hand gesture controller (front end)
4. Determine design of how to implement hand gesture controller.
5. Implement basic version of GUI
6. Implement the first hand gesture control
7. Test/debug basic version of application
8. Implement a more advanced version of GUI
9. Implement the other twelve hand gesture controls
10. Test more advanced version of application
11. Put finishing touches on application/allow time for unforeseen errors
12. Document everything we have done
13. Work on demonstration of our application

3.5 Schedule

We will be using an agile based development style, so some of these dates are not exact. We may be working on some of these tasks simultaneously, as they may not involve everyone. However, we plan to generally stick to this timeline to ensure all tasks are completed in a timely manner.

Tasks	Dates
1. Understand/gain background about using hand gesture input and controlling outside applications/software	11/14-11/28
2. Define exactly what actions/controls we plan to implement	11/29-12/17
3. Determine design of GUI for hand gesture controller	1/17-1/24
4. Determine design of how to implement controls.	1/24-1/31
5. Implement basic version of GUI	1/31-2/4
6. Implement firsthand gesture control	2/14-2/28
7.Test/debug basic version of application	2/28-3/7
8. Implement a more advanced version of GUI	3/7-3/21
9. Implement more hand gesture controls	3/21-4/4
10. Test more advanced version of application	4/4-4/11
11. Put final touches on application/allow time for unforeseen errors	4/11-4/15
12. Document the whole process	4/15-4/22
13.Work on final presentation/demonstration	4/22-4/29

3.6 Deliverables

- Design Document: Contains a listing of each major hardware and software component
    - Hardware includes: webcam (with model type), and computer used to run the developed application
    - Software includes: Linux distribution used and the developed application
- Python code for GUI application and TSM implementation
    - GUI application – OpenCV application in python which includes the wrapper class for integration of PowerPoint and a webcam.

    - TSM Implementation with modified gesture control functionality
- Presentation slides
- Final Report

4.0 Key Personnel

Andrew Parker – Parker is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Artificial Intelligence and Software engineering courses. He has experience in developing intelligent systems in the Unreal framework. He is responsible for bridging the TSM application to the usage of computer commands.

Daniel Miao – Miao is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Software Engineering courses. He has experience with AI prediction for medical patients as an intern at Arkansas State University. He will be responsible for assisting the bridging of the TSM application to computer commands.

Josh Stadtmueller – Stadtmueller is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Software Engineering and Artificial Intelligence courses. He has experience with developing front-end applications as an intern with Cobb-Vantress. He will assist in front-end development.

Garrett Bartlow – Bartlow is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Artificial Intelligence and Software engineering courses. He gained experience with developing back-end applications, cloud computing, and IoT devices while interning at Dover Fueling Solutions. He will be responsible primarily for implementing actions from hand gestures, however we will be taking a full stack approach. So, he will be involved in all aspects of the project. He will as act as team leader.

Jonathan Zamudio – Zamudio is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Artificial Intelligence and Software Engineering courses. He has experience with machine learning through the NACME Google Applied Machine Learning Intensive Summer 2021 Bootcamp. He will assist with bridging the TSM application to the computer commands.

Champion/Advisor name, Industry champion/professor – **Dr. Khoa Luu** received his Ph.D degree in Computer Science at Concordia University, Montreal City, Canada. His Ph.D. thesis was nominated for the Governor General Gold Medal in Canada. He was the valedictorian for the joint Faculty of Engineering & Computer Science and Faculty of Fine Arts convocation ceremony at Concordia University in 2014. His research interests focus on various topics, including Biometrics, Image Processing, Computer Vision, Machine Learning, Multifactor Analysis, Correlation Filters and Compressed Sensing.

5.0 Facilities and Equipment

The necessary equipment needed for this project is a linux operating system, a webcam, and a machine to run the linux operating system such as a computer at an on-campus computer lab.

6.0 References

[1] Lin, Gan, Han, "TSM: Temporal Shift Module for Efficient Video Understanding," Arxiv, MIT, 2019
[2]M. Elmahgiubi, M. Ennajar, N. Drawil and M. S. Elbuni, "Sign language translator and gesture recognition," 2015 Global Summit on Computer & Information Technology (GSCIT), 2015, pp. 1-6, doi: 10.1109/GSCIT.2015.7353332.
[3] T. Zhang, Z. Feng, Y. Su and F. Min, "Semantic Gesture Recognition Based on Cognitive Behavioral Model," 2014 International Conference on Information Science & Applications (ICISA), 2014, pp. 1-4, doi: 10.1109/ICISA.2014.6847463.
[4] M.V. Valueva, N.N. Nagornov, P.A. Lyakhov, G.V. Valuev, N.I. Chervyakov,
Application of the residue number system to reduce hardware costs of the convolutional neural network implementation,
Mathematics and Computers in Simulation,
Volume 177, 2020, Pages 232-243, ISSN 0378-4754.
[5] Tao, Wenjin & Lai, Ze-Hao & Leu, Ming & Yin, Zhaozheng. (2018). American Sign Language Alphabet Recognition Using Leap Motion Controller.
[6] Benitez-Garcia, Gibran et al. "Improving Real-Time Hand Gesture Recognition with Semantic Segmentation." *Sensors (Basel, Switzerland)* vol. 21,2 356. 7 Jan. 2021, doi:10.3390/s21020356