



Consortium for Computing Sciences in Colleges

Southeastern Region

**33rd
Annual
Southeastern
Conference**

**Student Research
Contest**

**Extended
Abstracts**

**October 25 and 26, 2019
Auburn University
Auburn, Alabama**

Table of Contents

1. Creating Interactive Fire Spread in VR
Andrew Webb – Roanoke College
2. Development of a College Retention Enhancement System using Machine Learning
Lee Phillips, Gilliean Lee and Quinton Amick – Lander University
3. Using Interactive Technology to Teach Basic Algorithmic Instructions
Josue Mayorga, Yassin Moghazy and Shawna Cassagnol – Georgia Gwinnett College
4. Using Technology to Create Synthetic Instruments and Engage Students While Teaching Algorithmic Skills
Alex Smyntyna, Khadijah Nixon, James Broome and Hieu Dinh – Georgia Gwinnett College
5. Reverse Engineering Malware: From Analyzing Assembly Code to Executing the Malicious Software
Daniel Haugen – University of North Georgia
6. Facial Emotion Classifier from Facial Expression using Deep Learning with Convolutional Neural Network
Aleksa Cupic and Gilliean Lee – Lander University
7. Developing and Evaluating a Usability Instrument for Mobile Augmented Reality
Derick Iem, Hunter Whitley, Elaicha Garcia and Nikki Mehdikhani – Georgia Gwinnett College
8. Skin Cancer Detection Using Convolutional Neural Networks
Mattia Galanti, Joshua John and Gilliean Lee – Lander University
9. We Put AR in a Room: Extending the Classroom by Unifying Independent AR-capable Platform to Produce Single-shared Experiences
Anas Doghaimat, Ephraim Kim, Jiawei Li and Daniil Nguyen – Georgia Gwinnett College
10. A Computational Approach to the Reconstruction of Proto-Japono-Koreanic
Alicia Key – Furman University
11. Automating Realistic Character Behavior
Bridget Kennerly and Ethan Kohrt – Furman University
12. Applying an Interactive Learning Style to Teach Programming Fundamentals Through a Familiar Activity
Daniel Redder, Joel Garcia, Nikki Mehdikhani and Patrick Page – Georgia Gwinnett College

Creating Interactive Fire Spread in VR

Author: Andrew Webb

Mentor: Dr. Anil Shende

1 Abstract

This research explored the methods needed to create a Virtual Reality (VR) training module for the United States Forest Service. The results gathered showed that while it is possible to take the Forest Service's simulation programs and process it into VR, there are limitations to it being web-based.

2 Introduction

The field of Virtual Reality, or VR, has been a steadily growing one. As it grows, VR has become a much more viable option in regard to education and training for fields that are often too dangerous or time and resource consuming to have proper training simulations. Some of these applications have been in the medical field, for instance: neurosurgery[1]. One field that has not been explored is the use of VR in order to help train firefighters in dealing with forest fires. Thus, in collaboration with the Fire Research Division of the United States Forest Service, my research focused on taking existing fire simulations and translating their results into VR.

3 Materials and Methods

The materials that were used involved an Oculus [3] Rift and the JavaScript based A-Frame [2], which is a script designed to allow for VR environments located in a web page. The overall design philosophy was to have the output given by the Forest Service's simulations, FARSITE [4] and STANDFIRE [5, 6], and have them processed by a Python program. This program would then create JavaScript arrays containing the data points for the terrain, tree, and fire spread for the given simulation. A web page containing A-Frame would then take these points and display them in a VR friendly web based environment. The need for the VR being web based was due to a desire to increase the scope of devices that could run the simulation.

4 Results and Discussion

The Forest Service provided test data in the form of a fire that spread amongst a 7.5 by 7.2km plot of land in Northern Montana. The Python program was able to successfully transfer the simulation data from the Forest Service to a format readable by A-Frame. As shown in Figure 1a, the terrain given is able to be processed and furthermore colored based off of the percentage of tree coverage per cell. Furthermore, the program can generate trees per cell equal to the amount of tree coverage as shown in Figure 1b. As well, the program read in fire data, and divided the spread into time slices, where the time slices are on a gradient from red to black, youngest to oldest. As shown in Figure 1c, the fire at time slice 20 out of 1000 is bright red. However, at time slice 258, the older fire begins to darken as shown in Figure 1d.

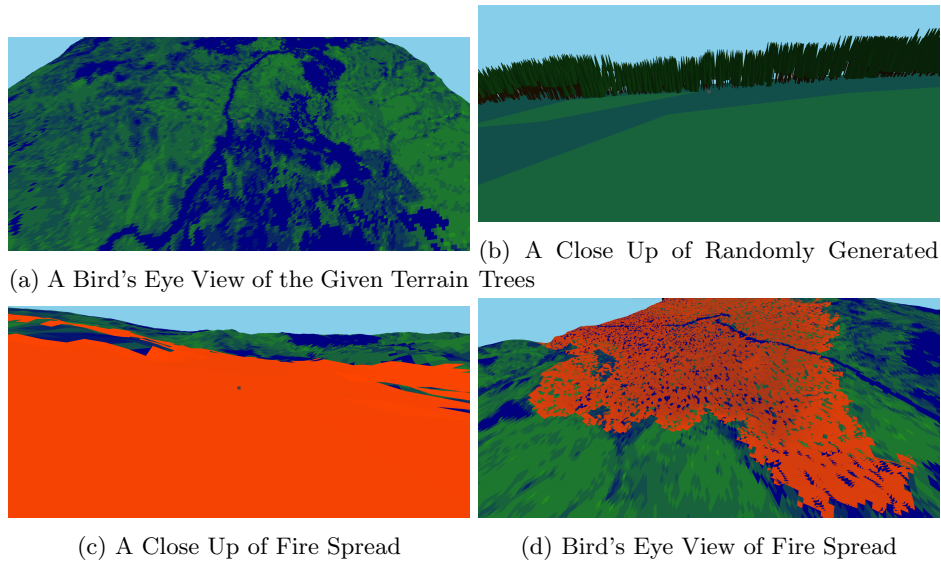


Figure 1: A Collection of VR Results

5 Conclusions

What the research has shown is that it is viable to take and process the output from the Forest Service simulation programs and transform it into a VR environment. However, while the process can handle displaying a simulation after it has finished running, FARSITE and STANDFIRE do not currently have a method to inject input into the simulation while it is running. This does limit the ability to create training modules without the option to update the simulation due to changes in the environment. Also, while the program can process the simulation, A-Frame has issues handling large amounts of faces and vertices. For more in depth simulations, a need to move to a dedicated engine would be required.

References

- [1] Alaraj, Ali, et al. *Virtual reality training in neurosurgery: Review of current status and future applications*. Surgical Neurology International, vol. 2, no. 1, 28 Apr. 2011
- [2] "A-Frame.". <https://aframe.io/>
- [3] Oculus.". <https://www.oculus.com/>
- [4] Mark A. Finney *FARSITE: Fire Area Simulator—Model Development and Evaluation*. Rocky Mountain Research Station Research Paper RMRS-RP-4 Revised March 1998, revised February 2004
- [5] Parsons, Russ, and Et. Al. "Modeling Fuels and Treatment Effects in 3D with STANDFIRE.". Advances in Forest Fire Research 2018, pp. 1384–1388., doi:10.14195/978-989-26-16-506-185.
- [6] Parsons, Russ, and Et. Al. "Modeling Fuels and Treatment Effects in 3D with STANDFIRE - Supplementary Materials". Advances in Forest Fire Research 2018, pp. 1384–1388., doi:10.14195/978-989-26-16-506-185.

Development of a College Student Retention Enhancement System using Machine Learning

Lee Phillips, Gilliean Lee, and Quinton Amick

Department of Mathematics and Computing

Lander University

320 Stanley Avenue, Greenwood, SC 29649

lee.phillips1@lander.edu, glee@lander.edu, quinton.amick@lander.edu

Introduction

Understanding student retention is important as university administrators try to improve graduation rates and student success as well as reduce the cost of recruitment by increasing the percentage of students who return after one year. With student data collected by a university, ML technology can be used to predict and classify a student's future with high precision. In an effort to increase retention at Lander University, the President's Decision-support Office, the Office of Institutional Research, and our research team collaborated to develop a Retention Enhancement System. After developing an input pipeline to preprocess student data so that it can be used for training and testing, we trained multiple machine learning models and measured performance as we continued to improve the system. The result is a system capable of identifying students at risk of not returning to the university with a strong degree of reliability. This system can be used to notify administration that a student may need support, as well as recognize areas a university needs to improve for retention purposes.

Development

Data Collection

Lander's Decision-support and Institutional Research Offices have provided the research team with 3 years of student data, containing almost 6500 student instances with more than 180 attributes. The last two years of data included extra attributes related to student concern and early alerts, such as absences, financial issues, substance abuse, etc. The data is highly confidential, so it was masked when provided to the research team.

Development of Input Pipeline

Input data has to be preprocessed so that training data supplied to ML system is uniform and can be used for efficient training. The preprocessing activities include identification and removal of meaningless and unimportant attributes, filling in missing values with median values, scaling of numeric feature values using standardization scaling method (scikit-learn developers, 2018a), and one-hot encoding of categorical data (scikit-learn developers, 2018b). Input data analysis and input pipeline development were performed using Python language with Scikit-Learn (scikit-learn, 2019) and NumPy (NumPy) packages. Plotting tools are used to visualize data to confirm or identify correlations, shown in Figure 1.

Training and Testing of Machine Learning System

Overfitting of the model is combatted during the first 2 phases by removing features, gathering more training data and reducing noise in the data. In this phase, overfitting is avoided through regularization (generalization) of the training model by adjusting hyperparameters, which are the parameters that impact how a model learns from the data and differ between models. Alternatively, underfitting is combatted by selecting more powerful models, providing better features to learn from and reducing the regularization from hyperparameters.

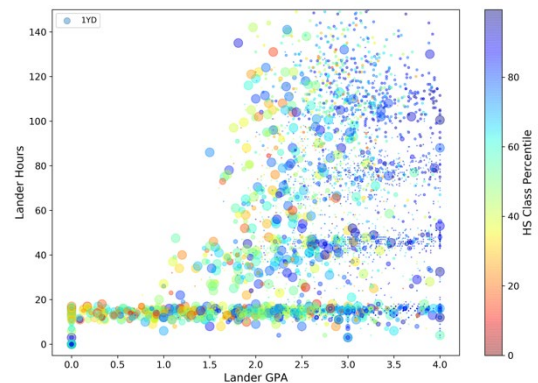


Figure 1. Scatter Plot of 1-Year Disposition, Lander Hours, Lander GPA, and HS Class Percentile

Stratified sampling was used to split the data 20/80 for training and testing, respectively. By encoding *1-Year Disposition* as a binary label, 0 meaning *no issue* (returned, graduated), and 1 meaning *issue* (transferred, suspended, dropped), we were able to simplify the model and increase performance. Due to the limited amount of training data, and particularly the low ratio of positive to negative samples, oversampling of the positive instances was used to improve the performance of the models.

Randomized Search Cross Validation and Grid Search Cross Validation were used to search for starting hyperparameter values and to fine tune model performance. We chose to train a Stochastic Gradient Descent (SGD) classifier, a Random Forest Classifier, a Support Vector Machine (SVM), and finally, a soft-voting classifier consisting of all three models. All of these models are available in the scikit-learn library (scikit-learn, 2019). The soft-voting classifier uses all three previously trained classifiers, along with each model's probability. As each model may weigh features differently in its decision process, this approach allows the strengths of the models to compensate for the weaknesses of others.

MODEL/ALGORITHM	HYPERPARAMETERS AND VALUES
Stochastic Gradient Descent	Learning rate = 'optimal', tol = 0.008, alpha = 0.01, loss = 'modified_huber'
Random Forests	Num of estimators = 1100, max leaf nodes = 400
Support Vector Machine	Polynomial kernel, coe0=0.6, C=1.9, tolerance=0.0008
Voting Classifier (ensemble)	Soft voting by SGD, Random Forests, and SVM

Table 1. Hyperparameters for Models

Classifier performance is compared using *Precision* and *Recall*, calculated from the confusion matrix, and the receiver operating characteristic (ROC) curve, as well as an F1 Score favoring recall over precision, shown in Figure 2.

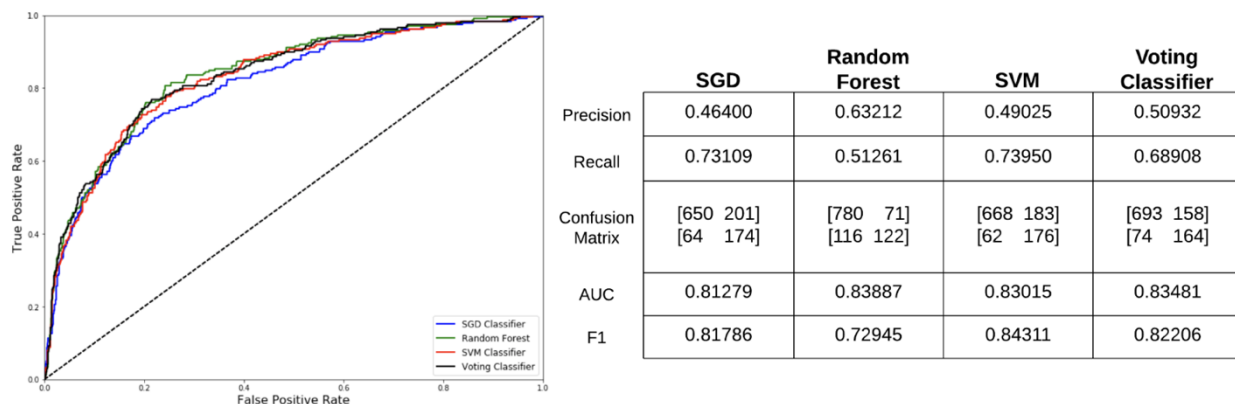


Figure 2. ROC Curve, Precision/Recall, F1 Score, and Confusion Matrix for Classifiers

Conclusion

The area under the curve (AUC) demonstrates the performance of each model, with an area of 1.0 representing a perfect model. Our models performed reasonably well with scores in the 0.80 - 0.84 range. The F1 score expresses the performance of each model, while also taking into account our desire to favor false positives over false negatives. By favoring recall over precision, we ensure that incorrect predictions tend towards false positives rather than false negatives, missing as few troubled students as possible, while maintaining accuracy >80%. This is where SVM stands out as the best performer, presently.

References

- Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn & TensorFlow*, O'Reilly Media
- NumPy developers. (2018). *NumPy*. Retrieved from <http://www.numpy.org/>
- scikit-learn. (2019, February 21). Retrieved from <https://scikit-learn.org/stable/>
- scikit-learn developers. (2018a). *sklearn.preprocessing.StandardScaler*, Retrieved from <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>
- scikit-learn developers. (2018b). *sklearn.preprocessing.OneHotEncoder*. Retrieved from <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html>

Using Interactive Technology to Teach Basic Algorithmic Instructions

Authors: Josue Mayorga, Yassin Moghazy, Shawna Cassagnol

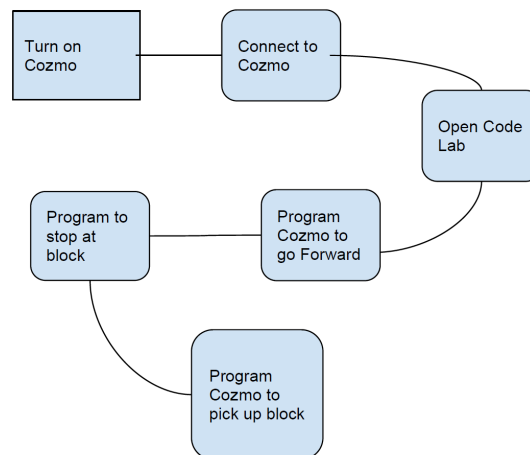
Mentors: Dr. Anca Doloc-Mihu and Dr. Cindy Roberston

School of Science and Technology, Georgia Gwinnett College

Interactive technology can be used to teach programming concepts to students more easily. Technology can seem hard to understand and scare off students with little understanding of it. The Technical Ambassador Program introduces some basic concepts through new and exciting technologies to show people that technology can be fun. The focus of the project is to teach students from ages 5 to 18 what an algorithm is and how it works while using interactive technology to pique their interest. The technology used is Cozmo: an advanced robot with an animated face and a voice that was created by Anki [1] (A Robotics Company). It can move around, play, and interact with people. Cozmo's brain is in the Cozmo App, which can be downloaded to your phone or tablet through the App Store and Google Play Store. The students will be asked to write a program that navigates a Cozmo inside a maze, retrieves a block, and returns it to the user.

Groups of students should not be less than 2 or greater than 4 people, as this is the most effective size for learning [2]. Algorithms can be easily explained by relating them to games students have already played, and therefore will be shown that they already understand the premise of algorithms which is set of instructions. After completing this portion of the lesson, they will be given access to, and specialized instruction for, Cozmo's programming software - Code Lab. When they are able to understand coding methods for this block-based programming software, they will be given access to the maze we provide and will have time to retrieve blocks from the maze using both the easy and intermediate versions of Code Lab.

Example of algorithm



Each maze will be set up differently and have their own challenges built in that the other mazes didn't have. This will force students to have to approach each maze differently than before but with the same fundamental concepts in mind. The goal is to get people who weren't interested in technology or have no idea how to approach it can use critical thinking skills to apply this to coding and realize that coding is just step by step instructions for a computer to process and execute. To our knowledge there are some example programs in the Cozmo app but not in a way to that teaches programming concepts like our project is doing.

Before using the Cozmo students will be asked to take a survey asking if they are interested in the ITEC field and how hard do they think it is. After the students have completed all activities they will be asked to take another survey. The survey will ask them how difficult they found the activities from a scale of 1 to 10. They will also be asked how enjoyable it was to use the Cozmo, as well if it changes their mind about the ITEC field and how interested they are in pursuing a career in it. The results of both surveys will then be used to determine if using Interactive technology can be used to teach programming concepts simpler and enjoyable.

References

1. Statt, Nick. "Anki's Cozmo robot is the new, adorable face of artificial intelligence" *The Verge*, 14 Oct. 2016' <https://www.theverge.com/2016/10/14/13276752/anki-cozmo-review-ai-robot-toy>
2. Ward, Beatrice A. "Instructional Grouping in the Classroom." *Instructional Grouping in the Classroom*, Education Northwest, 7 Aug. 2007, 1:03, <http://educationnorthwest.org/sites/default/files/InstructionalGrouping.pdf>

Using Technology to Create Synthetic Instruments and Engage Students While Teaching Algorithmic Skills

Authors: Alex Smyntyna, Khadijah Nixon, James Broome, Hieu Dinh

Mentors: Dr. Anca Doloc Mihiu and Dr. Cindy Robertson

Our team, in conjunction with the Technology Ambassadors Program (TAP), have been researching ways to improve technological literacy and comfort with computer code. By adopting Makey-Makey and Godot Technologies, we hope to create an interactive project that engages and teaches students. TAP is a school program that was created with the intention to provide students with a better understanding of technology and software. By showing students that programming can be practical and logical, we will encourage more people to strive for an IT major. As the mystery of programming and the links between the physical and computational realms shrink, we here at TAP aim to encourage interest in numerous technological fields.

Despite students in the world becoming more dependent on technology, there is a majority view on programming as intimidating and too difficult to understand. However, movements such as Hour of Code, which is still ongoing and has great outreach to teach students the basics of logical flow in coding, have proven that by exposing students to programming concepts through simple and engaging projects, they can let go of these misconceptions and open themselves to the world of programming and technology.

Our Synthetic Instrument project was created with a similar approach in mind while being far more interactive. A common issue when teaching students about programming is that programming doesn't feel practical or tactile; an easy way to improve the learning experience is to bring it into the physical world. Makey Makey, a simple circuit completing device which allows everyday objects to replace keyboard keys, allows students to interact with programs in basic but engaging ways. By allowing students to physically build an instrument using the kit and various conductive materials, they will have the creative freedom to design and play these instruments in whichever manner they choose. Each group of students will get a Makey Makey board, which has electrical inputs for wires. Copper wires, aluminum foil, coins, and metal pieces will be used to conduct the input from the students to the program in Godot which we developed to play sounds.

The Godot game engine, which uses GDScript- a programming language similar to Python- is what we will use to accept inputs from the synthetically created instruments then play sounds to the user. The engine is lightweight and will allow us to present easily to students how to manipulate the code and how to run it. By using simple visual elements to represent the movement of each note, the player will feel more in control of the sound. The Theremin is an instrument that was created relatively recently (1928); it is an odd instrument that uses the distance of the player's hand to the antennas to

produce sound in an analog way. We want to inspire students to think in creative ways of playing instruments, and producing sounds.

With the Makey Makey, our team aspires to bring the appeal of tech to be applicable to any individual. Students will learn how to utilize GDScript instructions by directly interacting with the code. This project is designed to teach algorithm skills to non-tech students in an engaging and interactive format. The main goal is to easily attract students to the wonders of technology. The Makey Makey allows for students to have a variety of choices in terms of keyboard inputs for the six sounds they choose to use. Each sound comes from prerecorded and prepared audio clips from different base instruments; students will have the ability to creatively mix and match these sounds to create their own unique instrument. They will also be able to increase and decrease the pitch of their chosen sounds.

Using an understanding of basic music theory and instrumentation we want students to create new and interesting ways to play instruments, working with shifting pitches and speeds using devices that have limited input capabilities. Students will be paired in groups of two or three and given a laptop, a makey-makey board, and a set of materials to create and wire a synthetic instrument. Students will receive a demonstration and instructions, which will guide them through the process of coding and building their instruments. We will guide students to programming and playing their own synthetic instrument, exposing them to how game engines run, and simple electrical engineering principles.

**Reverse Engineering Malware:
From Analyzing Assembly Code to Executing the Malicious Software**

Author: Daniel Haugen

Faculty Advisor: Dr. Bryson Payne

University of North Georgia

Abstract

Malware analysts and reverse engineers take software deemed to have malicious intent, analyze every aspect of the malicious code, and attempt to reclaim the confidentiality, integrity, and availability of the data previously infected with malware. Reverse engineering malware is the process of disassembling, debugging, and analyzing malicious software to understand its intent, determine how to remove it from a system, and hopefully prevent it from being installed onto another machine. This poster demonstrates the processes involved—including static, dynamic, behavioral, and automated analysis—in reverse engineering a piece of malware.

Within the static analysis stage, analysts must utilize disassemblers and decompilers to determine the purpose of the malware without executing the malicious code. Tools such as IDA pro and Ghidra will provide the necessary platforms for accomplishing most of the required steps within this stage. Although, applications such as PeStudio may also aid with static analysis. After that process, dynamic analysis involves running the software in an enclosed environment where the code is manually executed certain sections at a time. Utilizing debuggers like OllyDbg or WinDbg will allow analysts to view changes in memory caused by the program's execution and translation. The next level is behavioral analysis which encompasses allowing the program to run in a virtualized environment so that modifications made to the system can be recorded, providing examiners with a more realistic view of the malware in action. The final stage that can be taken to reverse engineer malware is the utilization of automated malware analysis sandboxes that will provide detailed reports regarding the nature of the malware. Utilizing these steps will allow this study to analyze a real piece of malware, document the implementation of the methodology referenced previously, and ultimately discern the intent and procedure employed by a piece of malicious software.

Classifying Human Emotions from Facial Expression using Deep Learning with Convolutional Neural Network

Aleksa Cupic, Gilliean Lee

Lander University, Greenwood, S.C. 29649, USA

Introduction

A good understanding of human emotions can have a profound impact on society and business. Our emotions influence every aspect of our lives, from how we connect with each other to how we make decisions, and to our health and well-being. Concerning the public sphere, government could benefit from detecting emotions such as guilt, fear, anger and uncertainty. TSA could benefit by scanning airline passengers for signs of terrorism, etc. in the process makes the world a safer place. Companies could also use emotion recognition to achieve competitive advantage. For example, emotion recognition could be used for hiring, measuring market's response and sentiment, and digital advertising. Businesses doing market research could determine how moviegoers are enjoying their movies. In advertising, emotion recognition could enable companies to connect with their audiences. Lastly businesses could use emotional recognition while interviewing people to see how enthusiastic people are about the position they are applying for.

Psychological research has classified six facial expressions which correspond to distinct universal emotions: disgust, sadness, happiness, fear, anger, and surprise [1]. Recently, Martinez discovered that there are 21 distinct emotion categories [2]. The research shows that happily surprised and angrily surprised are two different facial emotion expressions. We focus our study to detect four distinctive expressions (happy, angry, sad, and surprised).

Research Method

Convolutional Neural Network (CNN) is a very popular deep learning algorithm for computer vision [6]. Deep learning requires a large dataset to train and test. First of all, we have collected a large number of images with facial expressions from face databases such as 'AT&T'[3], 'Colorferet'[4], and around 800 hand-curated images from the Internet. Since the images doesn't come with emotion labels, we label each image manually, which is very tedious and time consuming. It can be hard to distinguish between happy and surprised sometimes.

In order to develop a classifier, we use TensorFlow framework [5] on Google Cloud Platform with the grant from Google Research Credits program.

Our model is a CNN with 3 layers, where each layer consist of Convolution (Conv2D), average pooling (AveragePooling), and Dropout layers with 0.2 dropout rate. In the first layer we have 2 Convolutions with 64 filters, in the second layer we have two Convolutions with 128 filter, and in the third layer we have one Convolution with 128. Kernel size is 3x3 in Convolution layers and 2x2 in Average Pooling layers. Also, we had 2 additional layers: Flatten layer to flatten our images into a single dimensional data, and Dropout. After Flatten we have dropout of 0.5, and after the first Dense layer we have another Dropout with 0.2 dropout rate to prevent overfitting. ReLU is used as the activation functions in those layers. The model has a Dense layer consisting of 512 neurons, and an output layer with 'softmax' activation function for multi-class classification as shown in *Figure 1*.

Data augmentation is a technique to use *artificially* generated images from each training image that are rotated, shifted, zoomed, horizontally flipped, horizontally and vertically shifted. The technique can help achieve higher image classification accuracy with smaller image dataset and avoid overfitting. During the training, we have used image augmentation with ImageDataGenerator from TensorFlow framework.

Results

We eventually used the image dataset with FER dataset from Kaggle [7], and RAF dataset [8]. We decided to use those datasets because they had many images and they were already annotated. We decided not to use the images we collected at the beginning of our research because the amount was insufficient. However, it taught us that cropping faces would improve accuracy significantly. *Figure 2* shows a few images from the test set and how cropping impacts accuracy.

FER and RAF datasets come with more than 4 emotions that we'd like to classify. We chose 19,146 training images and 2,052 validation images from FER dataset. All the images from FER database are grayscale and cropped. The problem with FER was that a number of pictures are mislabeled and have watermarks. On the original dataset we were able to achieve

79.19% validation accuracy. After going over every image to re-label manually, we were able to achieve a slightly higher validation accuracy of 81.7%.

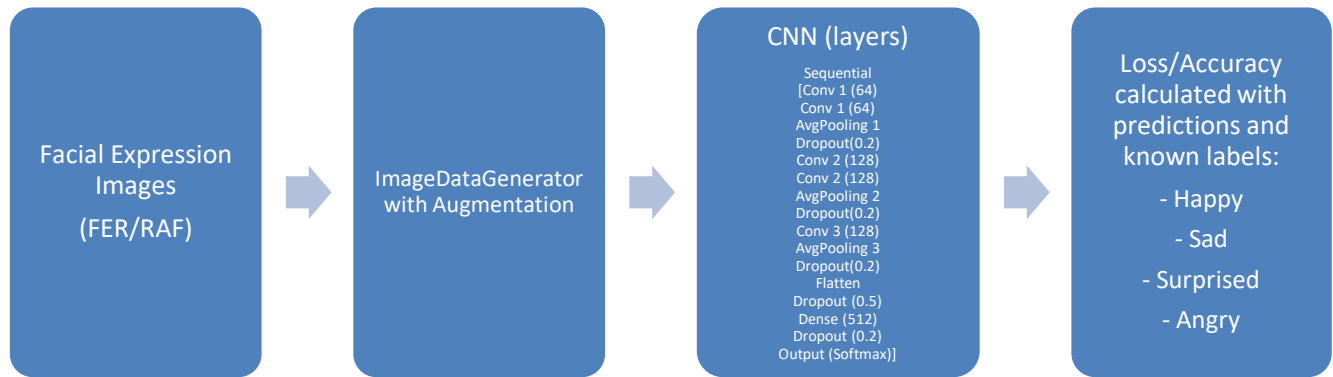


Figure 1: The Training Process of Facial Emotion Classifier

RAF dataset has cropped RGB images. We used 8,745 images for training and 2,154 images for validation. One defect of this dataset is that it is highly unbalanced. For example, it contained 3 times more of a happy facial expression images than others. After training, we were able to achieve validation accuracy for the two databases as shown in *Table 1*.

Table 1. Classifier Accuracy for Datasets

Dataset	Validation Accuracy Achieved
FER: 21,198 cropped grayscale images	81.73% (500 epochs)
RAF: 10,903 cropped RGB images	90.44% (300 epochs)

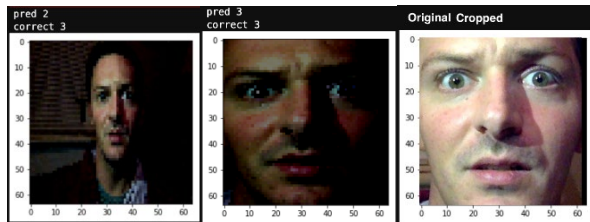


Figure 2. Uncropped image incorrectly predicted as 2 (sad). Cropped image correctly predicted as 3 (surprise).

References

- [1] Black, M. J., Yacoob, Y. (1995). Recognizing Facial Expressions in Image Sequences Using Local Parameterized Models of Image Motion. *International Journal of Computer Vision*. 25(1):23-48. DOI: 10.1023/A:1007977618277
- [2] Martinez, Aleix M. (2014/04/15). Compound facial expressions of emotion. , 111, E1454-.
- [3] Cambridge University Computer Laboratory, Database of Faces. (n.d.). Retrieved from <https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>
- [4] Flanagan, P. A. (2016, December 15). color FERET Database. Retrieved from <https://www.nist.gov/itl/iad/image-group/color-feret-database>
- [5] TensorFlow. (n.d.). Retrieved from <https://www.tensorflow.org/>
- [6] Saha, S. (2018, December 17). A Comprehensive Guide to Convolutional Neural Networks - the ELI5 way. Retrieved from <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- [7] Challenges in Representation Learning: Facial Expression Recognition Challenge. (n.d.). Retrieved from <https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/overview>.
- [8] Li, S., Deng, W., Du, J.: Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In: CVPR. pp. 2584–2593 (2017)

Developing and Evaluating a Usability Instrument for Mobile Augmented Reality Apps

Georgia Gwinnett College, Lawrenceville, GA 30043

Authors: Derick Iem, Hunter Whitley, Elaicha Garcia, Nikki Mehdikhani

Mentors: Derek Behmke and David Kerven

Abstract:

Augmented Reality (AR) is a relatively new technology that is branching out into the world in multiple ways. A problem currently facing mobile AR (MAR) applications is the lack of foundational usability metrics for app evaluation. This study will develop and test metrics that evaluate the usability of MAR apps in order to create a standard usability instrument by which MAR apps can be evaluated. The development of the usability instrument will leverage existing mobile app metrics and industry practices while still accounting for unique characteristics of MAR apps. Chosen metrics will be validated through testing with a variety of populations.

Introduction:

AR adds to the reality you would ordinarily see by overlaying additional digital information [1]. Effective AR applications can allow for a multi-dimensional understanding of complex data, faster navigation through convoluted spaces, and interactions with objects outside of current human imagination [2]. These upsides are balanced with potential negatives: a loss of situational awareness, the creation of cognitive overload, or even an interruption of workflow that can lead to poorer performance and a higher level of human error [2].

The iOS App Store has approximately 2,000 MAR apps available to consumers. The Google Play Store has another 200+ [3]. How many of these apps have been evaluated with usability metrics? Current industry professionals are implementing best practices kept from the basics of application usability and design principles, but they do not completely apply to MAR apps [4]. Evaluating MAR app usability has been a difficult task as there have been few studies done based on user experience with MAR applications, meaning that there are currently few methods on which to base an evaluation metric [5]. Consequently, this study aims to outline a general usability metric that may serve as a better tool to evaluate MAR applications.

Metric Development:

While many pre-existing usability metrics for mobile apps and general industry practices can be applied to MAR apps, these apps present unique aspects to consider such as the successful overlay of 3D artifacts, maintaining artifact location relative to the user, interaction with the real world, and filtering information relevant to the user [6]. Since MAR apps are mobile apps the following mobile app metrics will be included and tested: simple menu navigation, provided user support, ease of learning, and task completion time [7].

The International Organization for Standardization lays out three common standards for measuring usability: effectiveness, efficiency, and satisfaction [7]. These general guidelines will be used in tandem with the 6 P's as defined by the Toward Guidelines: present context-aware content, provide content controls, preempt technical difficulties, preserve intuitive icons and menus, promote social interactions, and pay attention to manipulability [8]. These two sets of principles, combined with the metrics in Figure 1, will be modified with an AR centric lens to generate an instrument that effectively evaluates the usability of MAR apps.

<u>AR Specific Metrics:</u>	<u>Mobile App Metrics:</u>
<ul style="list-style-type: none"> • 3D awareness • 3D overlaying • Ability to map surfaces properly • Maintain 3d artifact position in relative space. • Ability to interact with the real world • Ability to traverse 3D space • Environmental awareness 	<ul style="list-style-type: none"> • Effective use of the visual space • Simple menu navigation abilities • Provided user support • Features Present • Ease of learning • Response time • Completeness to tasks • Accuracy of task completion

Figure 1: AR and Mobile App Metrics

Methodology:

The proposed usability instrument will be piloted with three MAR applications: “Chemistry Simulator AR” created by Reptilesoft, “Augment - 3D Augmented Reality” by Augment, and an app created by students at Georgia Gwinnett College (GGC) that utilizes machine learning to recognize chemical molecules and display them in 3D space. Chemistry Simulator AR and Augment - 3D Augmented Reality are commercial applications that will be evaluated to ensure that the instrument and its metrics are valid not only for a more familiar application, but for others as well.

Participants in this study will be GGC professors and students. Pre-surveys will collect participant demographics and data to determine user comfort with technology. Next, participants will become familiar with the three apps in a tutorial phase, and then they will begin testing the usability of the apps. During testing, metric relevant data, such as time to complete objectives, will be collected. In the post survey users will rate their user experience and elaborate on the usability of the MAR app. Data collected will be analyzed for consistency to evaluate the proposed usability instrument. Upon successful validation of the usability instrument during this pilot, studies will extend to larger experimental population. The general process is outlined in Figure 2 below.

Participants In This Study

- Researchers, GGC Professors, and GGC students

Pre-Survey

- Age, gender, technological background, area of expertise, major, etc..

User-Testing

- Participants complete series of tasks while reporting on our usability metrics.
- Time and completion rate recorded.
- Audio and Video recordings for richer data set.

Post-Survey

- Participants will rate experience on a likert scale (1-7)
- Participants will be asked general questions about their experience using the AR apps

Analysis

- Results formed by participants using our metric will be evaluated.
- Audio-visual data will be reviewed and evaluated.
- Time for completion will be evaluated along side the results.

Validation and Improvement

- Results of analysis will help determine if the metrics used are valid.
- Inconsistencies will be addressed to improve the metric.
- Testing will re-occur based on the adjusted metric and expanded to more participants.

Figure 2: Experiment Workflow Diagram

References:

1. Emspak, Jesse. "What Is Augmented Reality?" *LiveScience*, Purch, 1 June 2018, www.livescience.com/34843-augmented-reality.html.
2. Endsley, Tristan C., et al. "Augmented Reality design heuristics: Designing for dynamic interactions." *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 61. No. 1. Sage CA: Los Angeles, CA: SAGE Publications, 2017.
3. Moon, Danny. "Mobile AR Is Evolving Faster than You Think." *VentureBeat*, VentureBeat, 25 May 2018, www.venturebeat.com/2018/05/19/mobile-ar-is-evolving-faster-than-you-think/.
4. Burkey, Scott. "XR Technologies for Industry 4.0." Georgia Gwinnett College. Cisco Auditorium, Lawrenceville, Georgia. 12 Sep. 2019.
5. Ko, Sang Min, Won Suk Chang, and Yong Gu Ji. "Usability principles for augmented reality applications in a smartphone environment." *International Journal of Human-Computer Interaction* 29.8 (2013): 501-515.
6. Livingston, Mark A., et al. *An augmented reality system for military operations in urban terrain*. NAVAL RESEARCH LAB WASHINGTON DC ADVANCED INFORMATION TECHNOLOGY BRANCH, 2002.
7. Lim, Kok Cheng, et al. "Usability Measures in Mobile-Based Augmented Reality Learning Applications: A Systematic Review." *Applied Sciences* 9.13 (2019): 2718.
8. SANTOS, Marc EC, et al. "Toward guidelines for designing Handheld Augmented Reality in Learning support." *Proceedings of the 23rd international conference on computers in education. china: Asia-pacific society for computers in education*. 2015.

Skin Cancer Detection Using Convolutional Neural Networks

Mattia Galanti, Joshua John, Gillican Lee

Department of Mathematics and Computing, Lander University, SC

Abstract:

Of all the types of cancer, a disease that causes more death than heart disease in wealthy countries [2], skin cancer is increasingly the most common in the United States [1]. In 2019, it is estimated that 7,230 deaths will be attributed to melanoma alone. However, skin cancer is also one of the most treatable types of cancer. The five-year survival rate for melanoma patients is 99% if it is detected and treated before it spreads to the lymph nodes, making early detection crucial. The warning signs of skin cancer include changes in size, shape, and color of moles or skin lesions [1]. The main objective of this research is to identify, given a certain image, the type of dermatological lesion or skin cancer.

Our research aims to develop a skin cancer classifier based on skin images using deep learning technology with TensorFlow [7]. We have developed classifiers on Google Cloud Platform (<https://cloud.google.com>) using Convolutional Neural Networks (CNNs) and image augmentation with a relatively small number of images to learn the technology. We were able to accomplish this due to the GCP Research Credits program, which granted us \$5000 to use for computing. We use CNNs, because they work well with any type of prediction problem involving image data as an input.

International Skin Imaging Collaboration (ISIC) has developed ISIC Archive, an international repository with more than 20,000 labeled dermoscopic images [3] [4] [5]. The images can be used to build a skin cancer classifier. Specifically, classifying dermoscopic images reach across nine different diagnostic categories: melanoma, melanocytic nevus, basal cell carcinoma, actinic keratosis, benign keratosis, dermatofibroma, vascular lesion, and squamous cell carcinoma [6], some of which are benign.

For this challenge, the research team built a dataset using several dermoscopic images, including ISIC Archive, and HAM 10000, known as Skin Cancer MNIST. The images from the ISIC archive were collected from clinical centers around the world and represent relevant clinical samples [8]. Dermoscopic images are images without skin reflex, from which a more in-depth view of the skin is obtained. Of these images we use 80% as a training set and the rest as a validation set.

The aim of the research is to develop a skin cancer classifier that predicts the diagnostic category from the image. We built two models from scratch – a binary classifier and a multiclass classifier – processing our data into our training and validation set. This was done by subdividing the various images in subfolders according to the type of skin cancer or lesion to which they belong. The images were then read by a data generator and converted to float32 tensors. Subsequently the images (normalized and with their labels) were fed to our network. For this step, we used the `ImageDataGenerator` class of Keras [9], which is a high-level API to build and train deep learning models. To prevent our model from overfitting and achieve higher accuracy, we used image data augmentation, which is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images [10]. In our project, we used some properties of the `ImageDataGenerator` class such as rescale, rotation, shear range and zoom range, that are used to augment images.

Our CNN models are composed of a Sequential layer that includes four convolutional/pooling layers, a Flatten layer that shapes 2d image data into single dimensional data, a Dense layer that is densely connected layers. There is a Dropout layer, which is a regularization technique to avoid overfitting in artificial neural networks, just before the dense layer. Adam optimizer that is known to converging to the global optimum quickly is one of the most popular optimizers in deep learning [11], and it is used for training our models. The last layer of the model is an output layer with softmax activation function for multi-class classifier or with sigmoid activation function for binary classifier, as shown in Figure 1.

With the test image data is ready, as we train the models with the data, we consider the loss and accuracy, which are indicators training process. Each time the training goes through the whole training data, we can see accuracy and loss. Accuracy is the portion of correct predictions, and loss is a measure of a model's performance. The model makes predictions off the training data, and then the prediction is compared against the known label, i.e. type of skin lesion/cancer. The collective results are used to calculate the accuracy. We use the validation set to compare the two obtained accuracies to correctly evaluate the results and see the performance of the classifier. Comparison of loss on the training set and validation set presents whether the model overfits or not.

We have developed two classifiers: a binary classifier that tells whether it is malignant or benign, and a multi-class classifier that tells the type of skin lesion/cancer. Our binary classifier's accuracy is about 78% with 150 epochs and our multiclass classifier's accuracy is about 72% with 200 epochs. Putting this into perspective, making a random guess at a diagnosis with 8 possibilities there is a 12.5% probability of guessing correctly, thus showing the potential of our models. The realization of this research project stems from the desire to contribute to the fight against skin cancer.

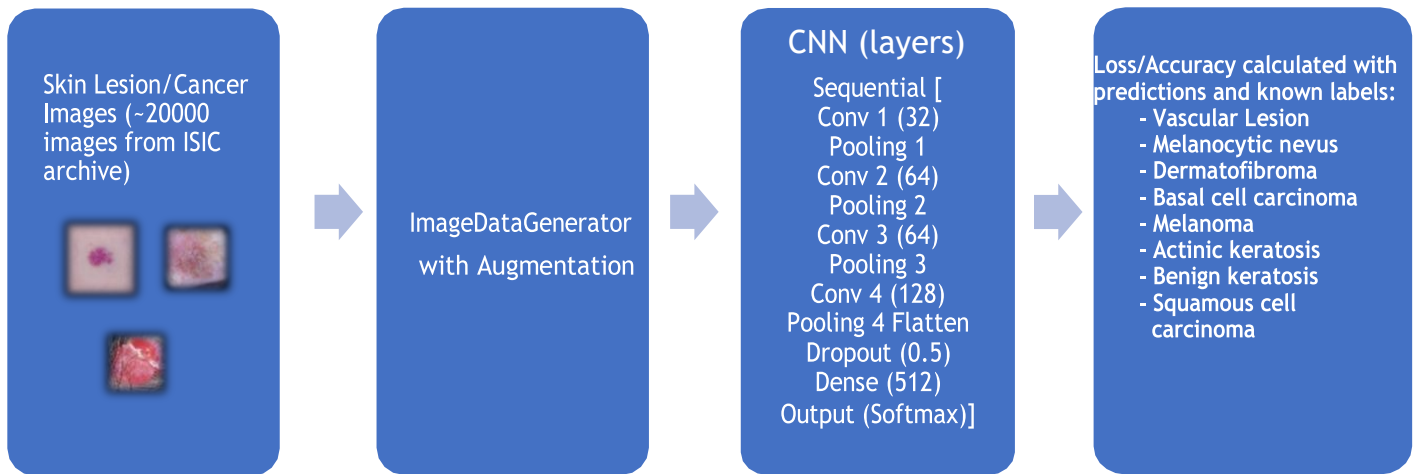


Figure 1: The Training Process of the Skin Cancer Classifier

References:

- [1] "Skin Cancer." Skin Cancer | American Academy of Dermatology, <https://www.aad.org/media/stats/conditions/skin-cancer>.
- [2] Blanchard, Sam, and Mailonline. "Cancer Is 'Overtaking Heart Disease as Wealthy Countries' Biggest Killer'." Daily Mail Online, Associated Newspapers, 3 Sept. 2019, <https://www.dailymail.co.uk/health/article-7421917/Cancer-overtaking-heart-disease-wealthy-countries-biggest-killer.html>.
- [3] Tschandl P., Rosendahl C. & Kittler H. The HAM10000 dataset, a large collection of multi- source dermoscopic images of common pigmented skin lesions. Sci. Data 5, 180161 doi.10.1038/sdata.2018.161 (2018)
- [4] Noel C. F. Codella, David Gutman, M. Emre Celebi, Brian Helba, Michael A. Marchetti, Stephen W. Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, Allan Halpern: "Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), Hosted by the International Skin Imaging Collaboration (ISIC)", 2017; arXiv:1710.05006.
- [5] Marc Combalia, Noel C. F. Codella, Veronica Rotemberg, Brian Helba, Veronica Vilaplana, Ofer Reiter, Allan C. Halpern, Susana Puig, Josep Malvehy: "BCN20000: Dermoscopic Lesions in the Wild", 2019; arXiv:1908.02288.
- [6] "ISIC 2019." ISIC 2019, <https://challenge2019.isic-archive.com/>.
- [7] "TensorFlow." TensorFlow, <https://www.tensorflow.org/>.
- [8] ISIC 2019, <https://challenge2019.isic-archive.com/background.html>.
- [9] "Image Preprocessing." Image Preprocessing - Keras Documentation, <https://keras.io/preprocessing/image/>.
- [10] "How to Configure Image Data Augmentation in Keras." Machine Learning Mastery, 5 July 2019, <https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/>.
- [11] Géron, Aurélien. Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly, 2018

We Put AR in A Room: Extending the Classroom by Unifying Independent AR-capable Platforms to Produce Single-shared Experiences

Georgia Gwinnett College

Lawrenceville, GA, 30043

Authors: Anas Doghaimat, Ephraim Kim, Jiawei Li, Daniil Nguyen

Mentors: Evelyn R. Brannock and Robert J. Lutz

Abstract

We intend to replicate an instructor-mediated, shared environment in a multi-platform capable Extended Reality (XR) application. XR can improve student learning of difficult content. Because it is intended for students, it should not exclude any students executing on a client such as: Android, iPhone, Magic Leap, HoloLens or Oculus Quest. Based on our research, a collaborative experience using disparate client devices does not exist nor was an underlying framework/API discovered. Therefore, we designed a prototype (XR) platform agnostic classroom environment. Teachers can control what the students see and experience while having their students follow along to digest and learn.

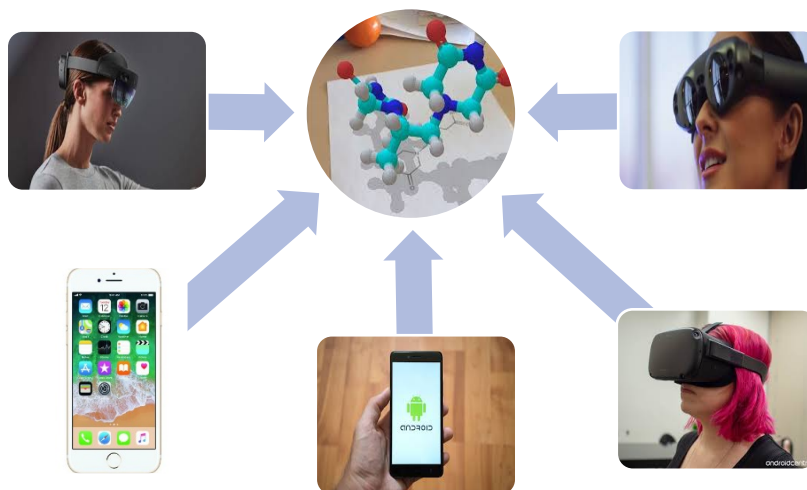


Figure 1: Android, iPhone, Magic Leap, HoloLens and Oculus Quest devices linked into the same room, all viewing the molecule that the professor presents and manipulates

Introduction

Extended reality (XR) technologies have made substantial progress in recent years, particularly in industrial and commercial applications where traditional training environments are being enhanced or replaced using XR technology (1). One environment that is especially promising for XR technology is the classroom, where conventional teaching and learning methods may be significantly elevated through the immersion that can be provided by XR technology. Two paradigms that dominate the classroom are the teacher-led model and the student collaboration model (2). XR technology can empower both paradigms by offering teachers a new way of presenting course materials and offering students an additional dimension for interacting with concepts in previously difficult or impractical ways.

State of the Art

There are a variety of interactive three-dimensional imaging software solutions that can potentially transform modern institutional environments (3). The zSpace classroom allows students to gather information by developing their own models and testing their ideas based on the models created. Students can then establish their unique conclusions based on their own ideas, and the data analyzed (4). Google Expeditions gives a very similar experience to young students by transforming a student's environment into whatever topic they are exploring (5). zSpace and Google Expeditions use kinesthetic realism that gives the user the ability to look around objects as if they were real.

Another early adopter of AR is the commercial manufacturing sector. AR technologies in a manufacturing environment can exponentially increase the numbers of employees training at a factory while increasing safety. Training costs are decreased drastically because the trainees are not using real equipment (6). Factory plant training employees use Mixed Reality (MR), which is the combination of augmented reality and virtual reality. One of the examples is Honeywell. Honeywell Skills Insight Immersive Competency aims for improving skills retention by 100 percent, reducing training time by 60 percent and preventing safety risks caused by training error. By implementing these technologies into the workforce, a company can reduce hazardous educational procedures.

MR also has an essential role in the medical field. ApoQlar Medical has had huge success in medical education and application. One of their main products is Virtual Surgery Intelligence (VSI), which combines all these technologies to deliver completely new medical solutions (7). For example, VSI helps trauma surgeons to locate fractures more precisely and easily.

Project Plan

In the previous iteration of this project, AR was brought to students by pairing augmented reality with machine learning to enable recognizing 2D structural patterns of molecules and constructing 3D versions of that molecule in the student's phone-based AR experience. We iterate on this work in the following ways.

Presently, our research found no collaborative platform that captures disparate client devices nor an application programming interface that unifies them. Our solution is to utilize a real-time database named Firebase from Google. This is accomplished by a process we called CastXR. The molecule from a single instructor device is delivered to multiple student client devices, thereby extending the teacher's manipulation and instruction of molecules to students. The implication for this capability in a classroom includes providing unique instructor-mediated experiences that enable students to follow along a mixed reality lecture, viewing classroom materials in a radically new way, and becoming more deeply engaged with the lecture material than they otherwise might be in a traditional classroom environment.

A professor first creates a room. Students are then able to enter this room and can spectate as the professor adds/modifies the desired molecule(s) in the room. Firebase provides automatic synchronized updates from the teacher's client to the database and from the database to the student's client. Only the teacher has the capability to edit and only the teacher's actions allow automatic propagation of the teacher's view to student's view.

References

1. Nechypurenko1, Pavlo P., et al. *Use of Augmented Reality in Chemistry Education*. ceur-ws.org/Vol-2257/paper02.pdf.
2. Getso, Muhammad, and Kinn Bakon. "Virtual Reality in Education: The Future of Learning." *ResearchGate*, Nov. 2017, www.researchgate.net/publication/323717453_VIRTUAL_REALITY_IN_EDUCATION_THE_FUTURE_OF_LEARNING.
3. Lee, Kangdon. "Augmented Reality in Education and Training." *ResearchGate*, Mar. 2012, www.researchgate.net/publication/257692981_Augmented_Reality_in_Education_and_Training.
4. Hite, Rebecca. "Learning in the Digital Age: A Review of the Research on Innovative Technologies." *ZSpace*, Oct. 2016, cdn.zspace.com/collateral/3DLearningWhitepaper_Oct_2016.pdf.
5. "Bring Your Lessons to Life with Expeditions | Google for Education." *Google*, Google, 2019, edu.google.com/products/vr-ar/expeditions/?modal_active=none.
6. King, John, and Joseph South. "Section I: Learning—Engaging and Empowering Learning through Technology." *Reimagining the Role of Technology in Education: 2017 National Education Technology Plan Update*, edited by Katrina Stevens, U.S. Department of Education, 2017, pp. 9–20.
7. "Virtual Surgery Intelligence (VSI) - The Smart Tool for Surgeons." *VSI.health*, Virtual Surgery Intelligence, 2019, vsi.health/en/.

A Computational Approach to the Reconstruction of Proto-Japano-Koreanic

Alicia Key

Supervisors: Alexander Francis-Ratte and Chris Alvin

Introduction

This research attempts to address the question of whether Japanese and Korean are genealogically related. Despite years of research into language families, linguists are still uncertain about the exact relation between Japanese and Korean. The Japonic-Koreanic question is important because it could potentially prove the existence of a new language family, reconstruct an ancient East Asian language, and contribute to a new chapter in the understanding of the evolution of languages. First and foremost, Japan and Korea are geographically close to each other, which makes it plausible that these languages are related due to proximity. Furthermore, both languages have the syntax of “Subject-Object-Verb”. Both languages utilize particles, many of which correspond to each other in usage. In addition, there are a fair amount of words that are similar in both languages, which is, in fact, how linguists attempt to prove that two languages are sister languages.

Cognates and Sound Correspondences

Building on the work of Francis-Ratte (2016), this research explored common words shared between the languages by comparing the sounds and meanings of words in the earliest available Japanese and Korean texts. This method enables researchers to trace cognates between the two languages and to reconstruct their common ancestor, ‘proto-Japano-Koreanic’. For our research we considered the sound correspondences shown in Table 1.

Table 1: Sound Correspondences Between Middle Korean and Old Japanese Implemented using our Framework

Consonants		Vowels	
p	p	a	a
t	t	i	i
k	k	wu	u
s	s		
m	m		
n	n		
l	r		

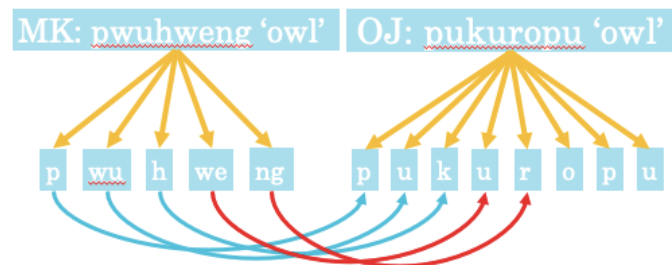


Fig. 1: String-Based Sound Correspondence Analysis of Middle Korean pwuhweng ‘owl’ and Old Japanese nukuropu ‘owl’

Computational Framework

We developed a program that calculates the number of attested sound correspondences in a list of Middle Korean and Old Japanese cognates. That number then becomes a strength rating that expresses how well that cognate fits in the proto-Japonic-Koreanic theory.

Our multi-step procedure is described by example referring to Fig. 1. The input is a sequence of Middle Korean and Old Japanese cognate pairs which we first break into sounds based on the word’s respective romanization. For Middle Korean, the words are written in Yale romanization while the Old Japanese words are written in the romanization used in Frellesvig (2010). Step two goes through each corresponding sound and checks to see if a sound correspondence occurs. For example, in Fig. 1, the first sound in the Middle Korean word, *p*, corresponds to the first sound in the Old Japanese word, *p* (as indicated with a lighter blue arrow). However, the fourth Middle Korean sound, *we*, does not correspond with the fourth Old Japanese sound, *u*. We track each attested sound correspondence maintaining a score

(based on count). In the case that the Middle Korean and Old Japanese words have different lengths, we loop until the end of the shortest word. This notion is important for the overall analysis of the cognates because the number of sounds that are actually analyzed will be used to determine an effectiveness percentage: number of attested correspondences divided by the number of sounds actually compared. In the case, of Fig. 1, we find an effectiveness percentage of 60% as 3 attested sound correspondences out of 5 analyzed sounds.

Simulation Results

The input data for this research consisted of a list of Middle Korean and Old Japanese cognates based on (Francis-Ratte, 2016). While the words have similar meanings, there is a portion of the list of cognate pairs would not support the proto-Japonic-Koreanic theory. Our goal is to determine computationally the best cognate pairs by sound correspondence and thus identify examples for the proto-Japonic-Koreanic theory.

Table 2: A Representative Sample of Output from our Implementation

Strong	Medium	Weak
MK: tatumicil ~ OJ: tatami Score: 5, 83.33%	MK: kalmo ~ OJ: karimo Score: 3, 60%	MK: kunul ~ OJ: kagey Score: 1, 20%
MK: twuti- ~ OJ: tuti Score: 4, 100%	MK: mesem ~ OJ: masura Score: 2, 40%	MK: hanolh ~ OJ: swora Score: 0, 0%

The scores and effectiveness percentages range from strong, medium, and weak, as evidenced by a sample set of output shown in Table 2: *A Representative Sample of Output from our Implementation* Table 2. The highest score from the tested list of Middle Korean (MK) and Old Japanese (OJ) cognates was 5 attested correspondences with the highest effectiveness percentage being 100%. The lowest score and effectiveness percentage is 0.

Future Work

While our initial implementation provides useful information, there is room for improvement to address richness of the linguistic analysis. For example, for Middle Korean and Old Japanese, it would be useful to add more sound correspondences with complex environments as well as add functionality to compare cognates that possess separable parts that are unhelpful for comparison.

Our implementation could be integrated into a larger system that would automate an overarching process taken by historical linguists. This mutli-phase process includes starts with the system providing expected *forms* of a cognate in a suspected sister language from either Old Japanese or Middle Korean. It would then search a dictionary for all words conforming to the expected form. For those words that are semantically equivalent (using human experts), the possible cognates can be rated by our system.

The novelty of our approach is that the computational framework is general enough from a linguistics perspective to compare any two languages, provided that the sound correspondences for said languages are identified and coded properly.

References

- Frellesvig, Bjarke. *A History of the Japanese Language*. Cambridge Univ. Press, 2010.
- Francis-Ratte, Alexander Takenobu, and J. Marshall Unger. *"Proto-Korean-Japanese: A New Reconstruction of the Common Origin of the Japanese and Korean Languages."* The Ohio State University, 2016.

Automating Realistic Character Behavior

Bridget Kennerly and Ethan Kohrt

Supervisor: Chris Alvin

Introduction

For any story to be engaging, the characters must evidence depth and behave in a realistic ways. Creating realistic characters is a major obstacle for modern videogames because non-player characters (NPCs) either have to be hardcoded or generated in-game using a template. Both methods allow for little variability in character behavior. Our goal is to generate nuanced characters that can more realistically interact with player characters, other NPCs, and their environment.

Motivating Character Factors: Values and Emotions

Humans and their behaviors are complex; we make decisions based on a personal set of values, but also act on our emotions. Although complex, we attempted to model this complexity in our own characters with the intent that their actions would evidence similar nuanced behavior. Our character model consists of two main components: values and emotions.

Values are abstract, assist in generating goals, and motivate actions. This sequence is malleable to new and changing environments and thus characters can react in any situation. Values represent categories of things that a character would like to have happen; for example, a character who values Hedonism will tend towards actions that increase pleasure, or a character who values Security will tend to play it safe. To guide our model we used the Schwartz Theory of Basic Values, which posits ten basic human values: Hedonism, Stimulation, Self-Direction, Universalism, Benevolence, Conformity, Tradition, Security, Power, and Achievement [3].

Emotions on the other hand offer reactions to specific kinds of stimuli, such as danger, novelty, gain, loss, or an encounter with a friend or enemy. To model emotions we used Robert Plutchik's theory of emotions, which proposes eight basic emotions: Joy, Trust, Fear, Surprise, Sadness, Disgust, Anger, and Interest [2]. The interplay of a character's emotions and value set allows them to make nuanced decisions in their environment.

Environment

A simple environment was needed in which to study our character models. This environment is represented as a grid, in which two kinds of entities exist: assets and characters. Assets represent inanimate objects (e.g., a vase, a table, etc.) located on the grid and are given a set of qualities that defines how a character may interact with them (e.g., i.e. "holdable," "edible," etc.).

Characters are much more complex: they have a set of values, with each in the range $[0,1]$ representing how much the character values it (0 valued not at all). They have a basic personality, which determines how sensitive they are to each of the eight emotions; they have an object called a Mind Graph [1], which tracks the character's relationships and emotions towards each entity they have encountered, and they also have health, stats, and a location on the grid. Because every aspect of a character is represented numerically, it is easy to randomly generate new characters that behave in different ways.

Generating Actions

Given an environment populated with characters and assets, characters perform actions using a turn-based approach. The choice of an action for a character is simple: it consist of an allowable random action. However, what actions are possible in an environment for a character? We generate the set of possible actions using a template-based approach that is populated at runtime. That is, we begin with a predefined set of action templates with placeholders for specific characters and assets. These actions each have

conditions which must be met in order for the action to be possible: factors like distance, stats, qualities of assets, and many others are taken into account. As an example, a character must move toward an item before it may interact directly with it (e.g. pick it up). For a particular environment, the set of all possible actions is generated each turn for each character; once an action is chosen, it is executed.

Choosing Actions

Characters choose actions that align with their value set and with their current emotions. Every generated action is evaluated by performing the action within a duplicate environment – this way the effects of the action can be analyzed before actually performing it – and based on the effects of the action it is labeled with its own values and emotions, which can then compared to the character’s own values and emotions. The more closely they are aligned, the more likely the character is to choose that action.

Emotional Stimuli

One character’s actions sometimes trigger an emotional response in a second character. When this happens, the second character’s emotions towards the first are updated, and this character becomes more likely to act emotionally towards the first character. All emotions die down over time.

Figure 1: Sample Output of Character Interactions in our System

Keanu picked up the cake	Meredith entertained Keanu	Meredith punched Keanu
Meredith studied the medicine	Keanu became joyful	Keanu became afraid of Meredith
Keanu shoved Meredith	Keanu entertained Meredith	Keanu punched Meredith
Meredith picked up the medicine	Meredith became joyful	Meredith became angry at Keanu
Keanu threw the cake at Meredith	Meredith walked east	Meredith punched Keanu
Meredith gave the medicine to Keanu	Keanu walked east	Keanu died.
Keanu examined Meredith	Meredith ate the cake	
Meredith studied the cake	Meredith became joyful	
	Meredith became sad	

Results and Future Work

Our system implementation runs for a given number of turns and outputs a chronological series of strings explaining which actions each character performed. It also gives a description of any emotion changes in a character after each action. For example, Figure 1 shows output from three separate runs (one per column) in an environment that contains two characters, Meredith and Keanu, as well as two Assets, cake and medicine. The left column demonstrates the concept of action conditions, in that the character Keanu is unable to throw the cake before he has picked it up. The second and third columns are from runs where emotional reactions are turned on; the second image shows that Meredith becomes happy after eating the cake, but then is sad since she no longer has the cake.

In the future it would be useful to include which values and emotions most strongly influenced the character’s choice of action; due to time constraints this feature was left out. Another limitation comes from the turn-based model: since our system does not represent time in a realistic way, a more nuanced model could provide better results, for example, giving each action a variable duration.

References

- [1] M.J. Alvarez, R.E. Amaya, et. al. “Hello Narratives! Character Development in Automated Narrative Generation. *Florida Artificial Intelligence Research Society Conference (FLAIRS)* (2019): 265.
- [2] Plutchik, Robert. “The Nature of Emotions.” *American Scientist* 89.4 (2001): 344-350. Web. JSTOR.org (17 June 2019).
- [3] Schwartz, S. H. “An Overview of the Schwartz Theory of Basic Values.” *Online Readings in Psychology and Culture* 2.1 (2012). Web. doi.org (20 June 2019).

Applying an interactive learning style to teach Programming

Fundamentals through a familiar activity

Authors: Daniel Redder, Joel Garcia, Nikki Mehdikhani, Patrick Page

Mentors: Anca Doloc-Mihu, Dr. Cindy Roberston

School of Science and Technology, Georgia Gwinnett College

Information technology is the fastest growing industry reaching throughout all fields of study. Gathering information and knowledge is essential for an organization to succeed. Using the gathered knowledge, an organization can apply and adapt to the fast-paced internet-based society. Whether an organization is able to effectively use this data will either make or break them. The main boundary potential Information Technology students face is a belief that programming is an incredibly difficult skill to learn. This misconception is perpetuated by movies and other forms of media displaying programming as a “how fast can you type” simulator, however in reality programming is a methodical process to achieve a result, which is what led us to the idea of using a sport to teach programming. We believe that using a device such as Sphero will help to break down such misconceptions and will facilitate faster learning.

We are enrolled in the Technology Ambassador Program (TAP) with the goal to develop an interactive workshop to help middle school to college non-techy students to learn basic programming concepts. TAP will provide us with the necessary tools to build the obstacle course and resources (Spheros) in order to develop our interactive workshop along with mentoring to ensure the best possible outcome for the study.

We are using Sphero as a tool to teach basic programming concepts to students of varying technological expertise. Sphero is a programmable robot with both a text based and block based interface. When learning something new it is helpful to have something known, so we are using Sphero to create a familiar environment to learn programming. By emulating a sport, in this case basketball, we are providing a physical context for our activities.

Our workshop will take place in a classroom setting. We will introduce our Sphero project with a brief PowerPoint presentation and a showing of Sphero's capabilities in a controlled demonstration. Students will learn the task of navigating the Sphero through a basketball themed obstacle course. Then, students will complete various basketball "drills" like driving through cones, following patterns, and "scoring a goal" throughout the course. TAP members leading the workshop will split up to assist the groups in a Sphero basketball-type challenge. We will add a competitive edge by allowing students to compete for the fastest time navigating through the course.

We will conduct a small assessment of their programming knowledge before and after the workshop as well as a survey based on the course's shortcomings and its strengths. We will use the data obtained from these assessments to visualize any improvements the students achieved. The surveys will be used to fine tune our workshop routine, measure whether we were effective in reaching our goal, and to further streamline the learning process.

We will be running this workshop mainly to direct college students towards STEM classes, however we will have participants ranging from middle school through college age. We will create different tasks for different age levels so that all participants can remain engaged and encouraged to participate. With all age groups we will use the block based interface provided by Sphero; the difference between the groups will come from content. With the college students we plan on teaching them the meaning of the major basic programming concepts: loops, if-statements, and functions. With the younger groups we will focus more on giving them exposure to the technology, and a general understanding of what coding is.

Our goal is that after our activity the students will develop an interest in STEM fields while learning basic programming concepts. We want the activity to show students the interactive side of programming and some of its capabilities. Beyond the workshops and lessons we hope to prove that the addition of interactive lessons in beginner level courses can solidify basic concepts of the subject matter at hand.