



Xuanyi (Steven) Zhu

A valuable, forward-thinking, and versatile computer science master student from University of Illinois at Urbana-Champaign. Proficient at software development, algorithms, ML and full stack web applications.

Seeking full-time software engineer positions.

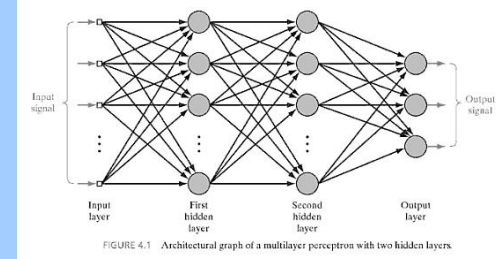
(217) 419-6173 xzhu42@illinois.edu

www.linkedin.com/in/xuanyi-zhu

<https://github.com/XuanyiZ>



- Used Python and Tensorflow to build neural network(MLP, CNN, LSTM) models that classify noise and human sound to help the audio team achieve noise reduction/cancellation goal with 97% accuracy.
- Applied normalization and regularization with optimal parameters to overcome overfitting issue.
- Optimized and evaluated performance via feature selection, k-fold cross-validation, recall, and precision.
- Presented analysis results to executives. Wrote an ML/AI concept and resources tutorial book.
- Gave lectures to the engineers to help them quickly grasp ML knowledge.



10ms as a frame, 20ms
analysis window with 50%
overlapping, 512 point FFT,
fs: 16,000 Hz



Meet Happy.
Zoom Video Communications



ArcSoft market inventory management system

A full stack web app that helps track and analyze current market data and reduce labor/time costs.

Email

Password

Log in!

Designed and Developed by Xuanyi (2018)

Dashboard

0 Total Sales Quarter

90 Total Transactions

33 Products

LowLightShot in 2018

Product Transaction Table

Competition Table

Market Product Matrix Chart

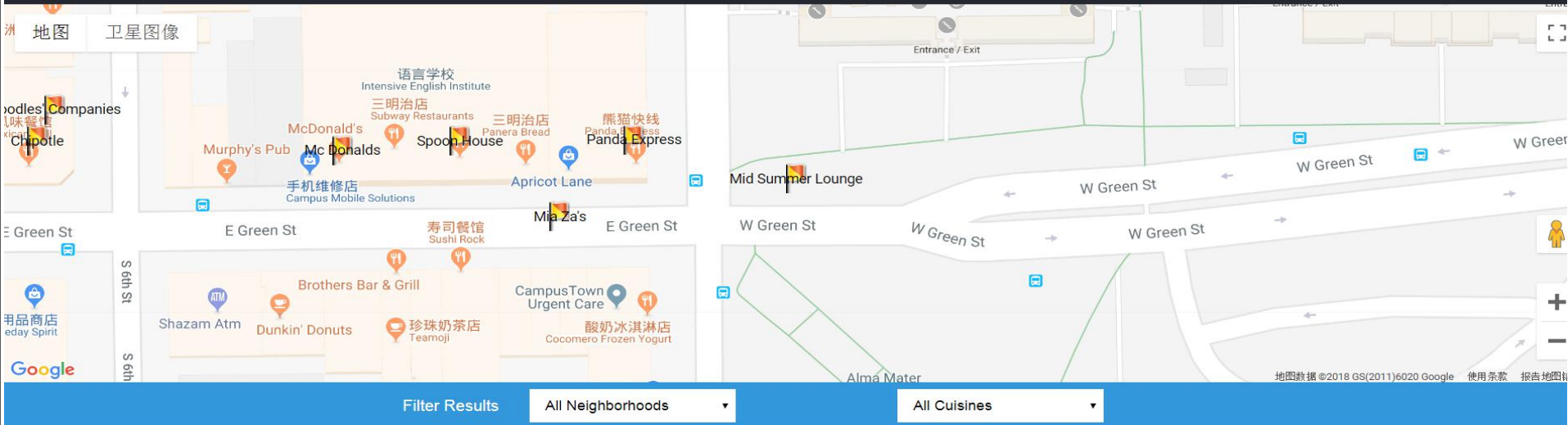
Skills: PHP, JavaScript, JQuery, AJAX, HTML, SQL, Bootstrap, CodeIgniter, HighCharts

— Eattogether

A web app that leveraged busy students' lunchtime to facilitate information exchange and social circle expansion.



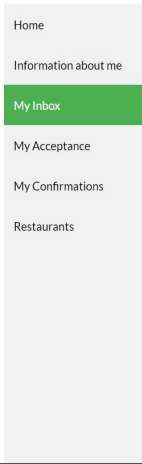
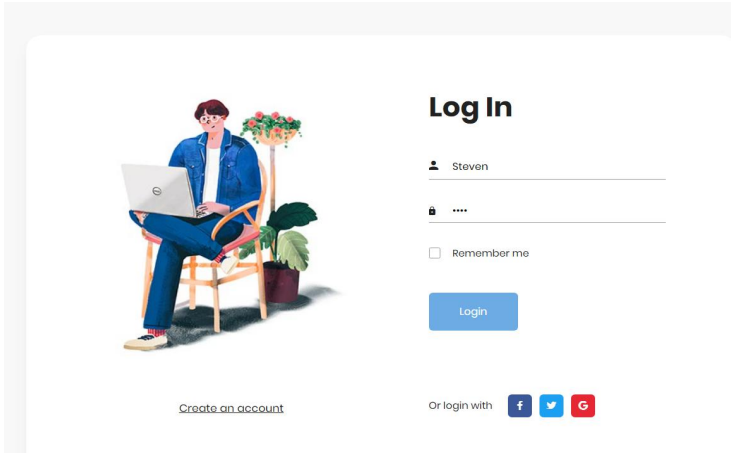
RESTAURANTS



Available: <http://eattogether.pythonanywhere.com/>

— Eattogether

A web app that leveraged busy students' lunchtime to facilitate information exchange and social circle expansion.

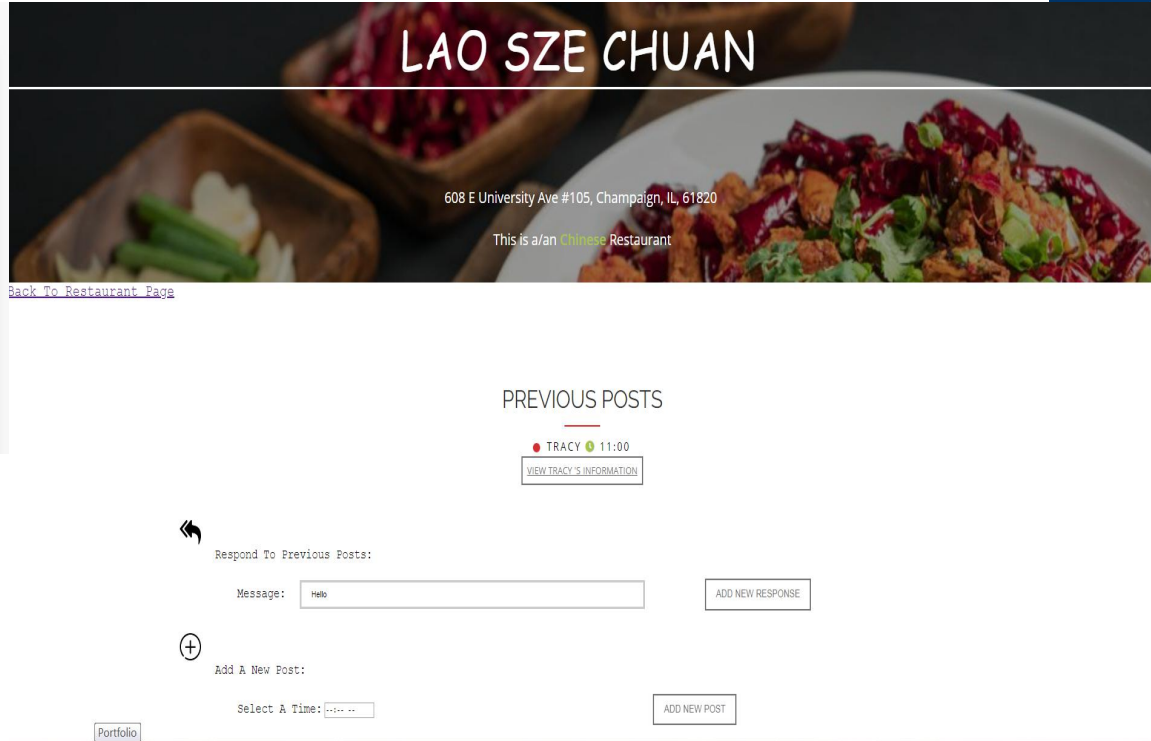


MY INBOX

tracy Sakanaya 20:08

● STEVEN ● PLS DINE WITH ME:) LETS CHAT!

SUBMIT



Skills:

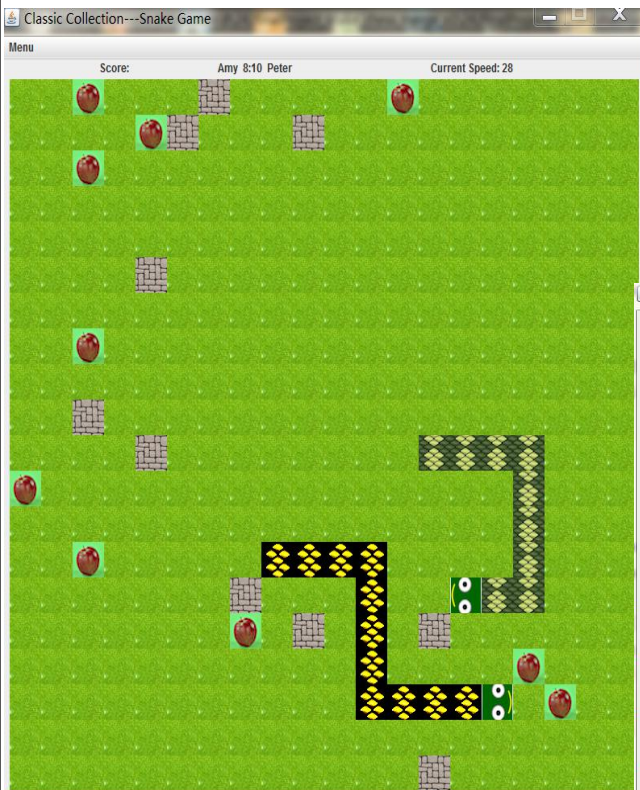
Python, Flask, MongoDB, Google map API, cloud, RESTful

Available: <http://eattogether.pythonanywhere.com/>



— Classic-Game-Collection desktop application

An interactive classic game collection including Snake, Chess, and Sudoku.

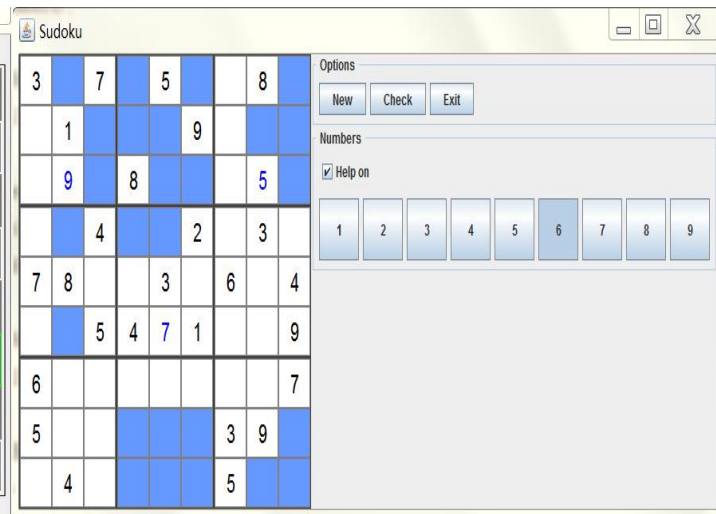
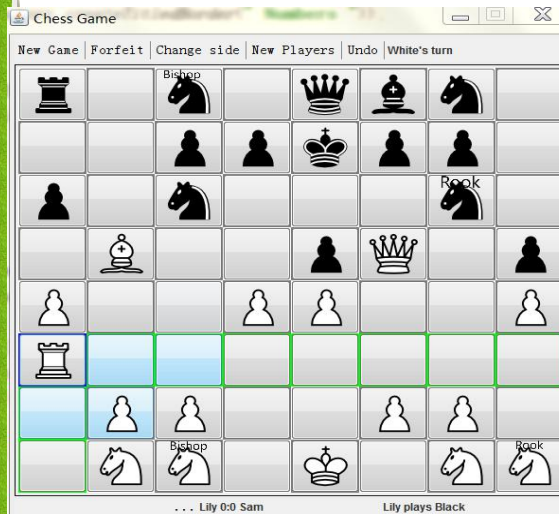


Target users:

Patients: recover cognitive consistency and psychological balance

Children: for gaming enlightenment

All age ranges: who loves to play classic games.



Skills: Java, JUnit, Swing dynamic GUI, MVC, OOD

Available: <https://github.com/XuanyiZ/Classical-Game-Collection>



— Twimalizer desktop app

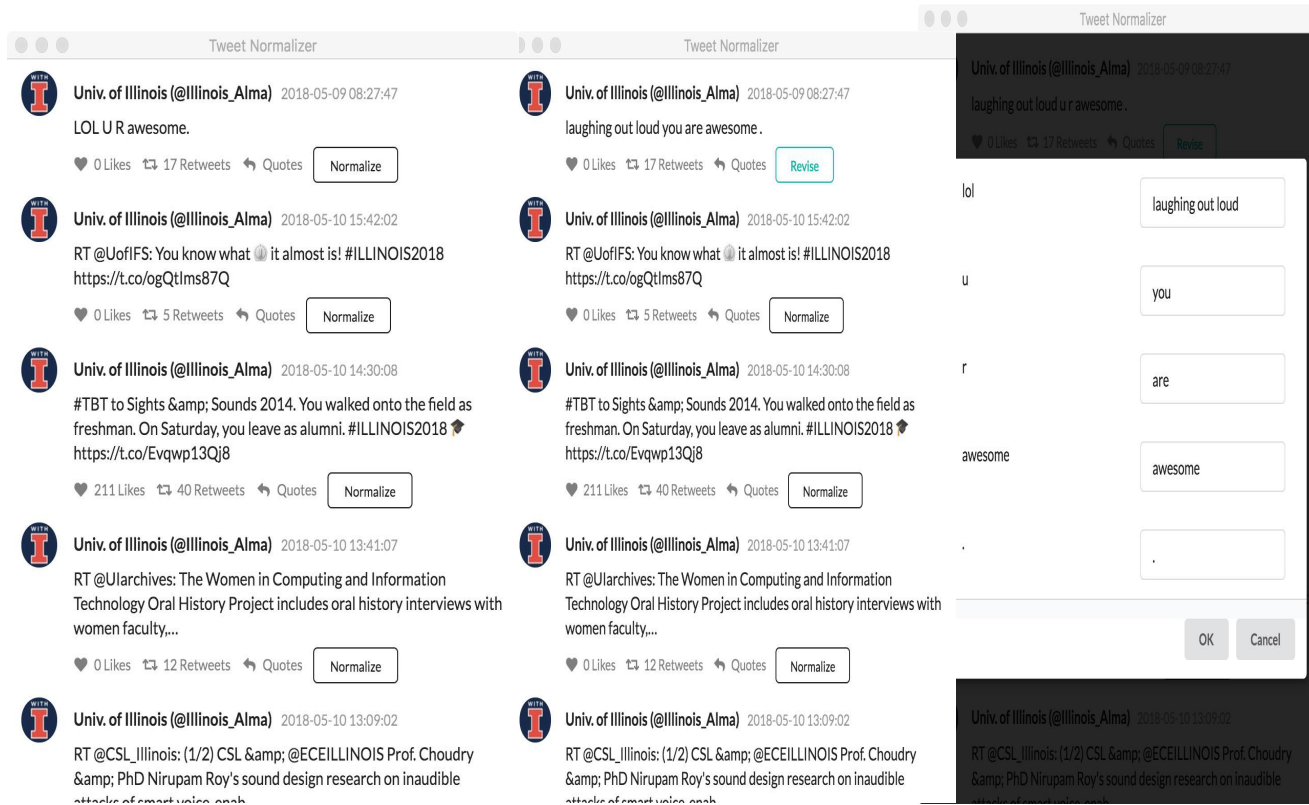
A supervised-machine-learning system to perform lexical normalization for English Twitter text.

Key Features:

- Generated candidates based on past knowledge and a novel string similarity measurement.

- Used Electron and Vue.js to implement a GUI which interacts real-time with the Twitter API, parsing and displaying the data in the application interface.

- Enhanced accuracy by supplying user-aided revision features that enable normalization engine evolution.



— NextGreatChef

A recipe sharer web application that helps users to learn and share different recipes worldwide.



My Profile

Home All Recipes Logout

Welcome! nicole111

*Your basic information:

Name: nicole

Email address: nicole@gmail.com

Preferred World Cuisine Type: Italian

*Recipes you added before:

Key Features:

Recipe recommendation: based on users' past view history and desired intake of calories.

Map Of Favorites: a world map shows users' favorite recipes based on the Google map API.



*Recipes you liked:

Italian Roast Beef

Addictive Sesame Chicken

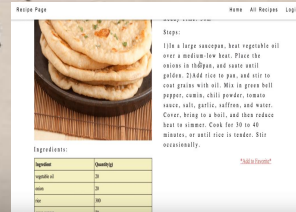
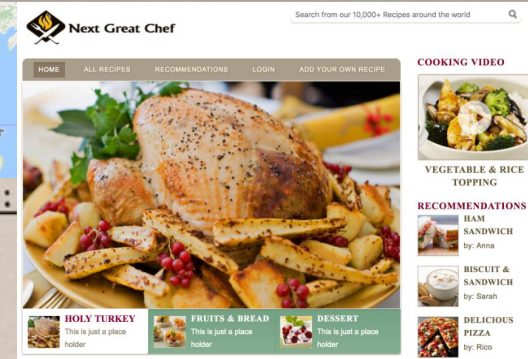
Mexican Rice

Japanese Onion Soup

Chicken Katsu

*We can recommend some recipes to you based on the maximum calories intake you want

Please enter your maximum intake of calories:

 Search

Skills: PHP, SQL, MySQL, HTML, CSS, Javascript, Google Analytics

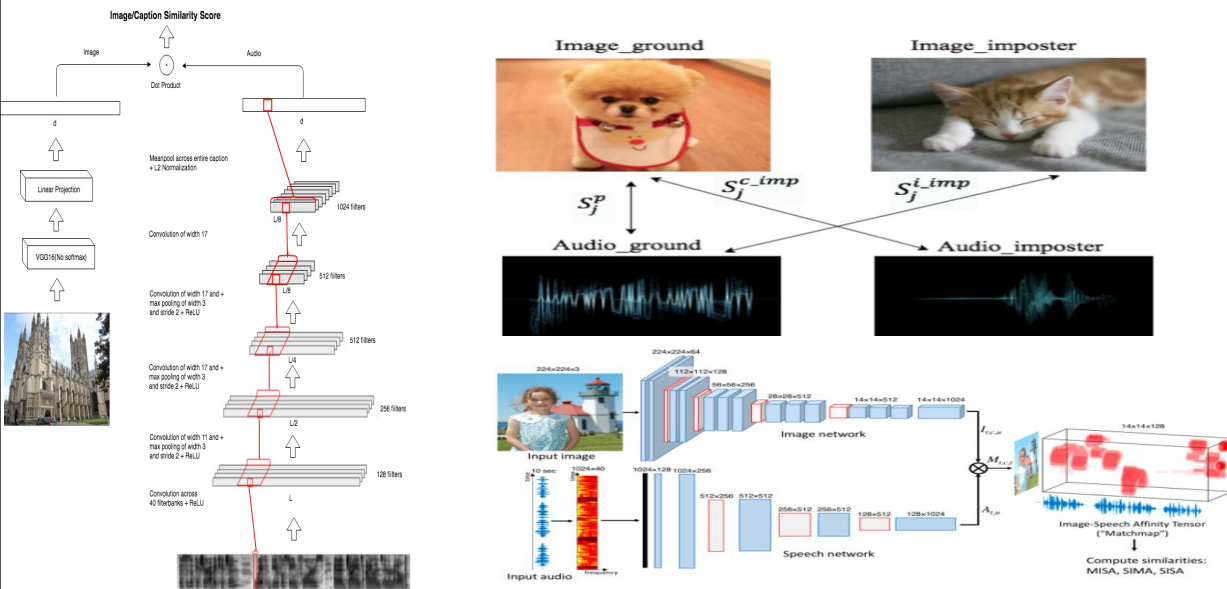
Available: <https://www.youtube.com/watch?v=cMZktza4BSw>
<https://github.com/Xuanyiz/nextGreatChef>

Optimization Of Multimodal Analysis Of Image And Audio Pairs



Explored the learning ability of neural nets for Image And Audio Pairs Matching.

Research about neural nets discovering word-like acoustic units from continuous speech at the waveform level with no additional text transcriptions or conventional speech recognition apparatus and grounding them to semantically relevant image regions.



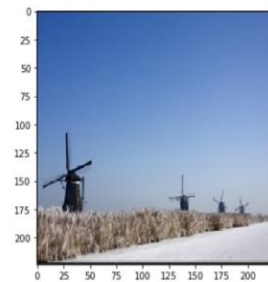
Ground Truth Image:



Image Annotation

Ground Truth ASR Text:

picture taken at sunset over looking into windmills in a field



a wheat field with many windmills lined up in it there is a clear blue sky in the background

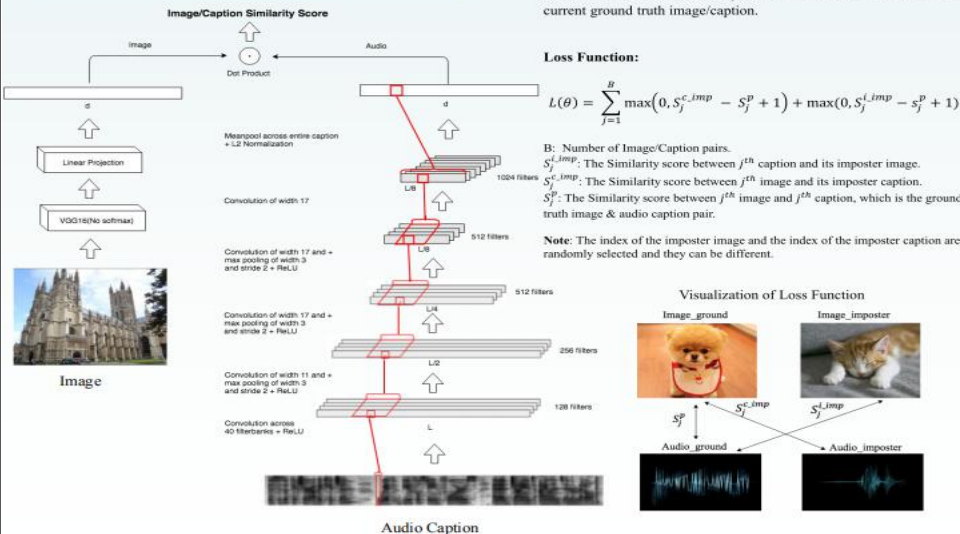
Optimization Of Multimodal Analysis Of Image And Audio Pairs



Abstract

Given a dataset of images and spoken audio captions, the paper written by Harwath et al. (2016) presents a method for discovering word-like acoustic units from continuous speech at the waveform level with no additional text transcriptions or conventional speech recognition apparatus and grounding them to semantically relevant image regions. Based on their work, we extend and optimize their model to achieve better performance and accuracy to find a better match between image and audios. And our optimized machine learning algorithm can be used to search images based on audio captions or search things in the opposite way.

Model Architecture



Baseline Model

Image Branch:

Taking VGG 16 layer network (weights fixed) and replacing the softmax classification layer with a linear transform which maps the 4096-dimensional activations of the second fully connected layer into our 1024-dimensional multimodal embedding space.

Audio Caption Branch:

Transform the audio segments into speech spectrograms and feed into a customized convolutional neural network as shown below.

Connection Part:

The overall multimodal network is formed by tying together the learned visual feature vectors and learned audio feature vectors and map them into a shared embedding space. Then, computes an inner product between them, representing the similarity score between a given image/caption pair.

Terms:

Imposter Image/Caption: An image/caption that is randomly selected from the current batch, but is its content is different from the current ground truth image/caption.

Loss Function:

$$L(\theta) = \sum_{j=1}^B \max(0, S_j^{c, imp} - S_j^p + 1) + \max(0, S_j^{i, imp} - S_j^p + 1)$$

B: Number of Image/Caption pairs.

$S_j^{c, imp}$: The Similarity score between i^{th} caption and its imposter image.

$S_j^{i, imp}$: The Similarity score between j^{th} image and its imposter caption.

S_j^p : The Similarity score between j^{th} image and j^{th} caption, which is the ground truth image & audio caption pair.

Note: The index of the imposter image and the index of the imposter caption are randomly selected and they can be different.

Optimization

Modification of Loss Function

The basic idea is inspired by the original loss function. But we will find the matched image for a selected imposter caption and find the matched caption for a selected image for calculating loss function. And we modified the loss function to be following:

$$L(\theta) = \sum_{j=1}^B t1 + t2 + t3$$

B: Number of Image/Caption pairs.

$$t1 = \max(0, S_j^{c, imp2image} - S_j^{c, caption_gimage} + 1)$$

$$t2 = \max(0, S_j^{c, caption_imp1image} - S_j^{c, caption_gimage} + 1)$$

$$t3 = \max(0, S_j^{c, caption_imp2image} - S_j^{c, caption_gimage} + 1)$$

$S_j^{c, caption_gimage}$: The Similarity score between A's caption and B's image when we are using j^{th} image/caption pair as the ground truth image/caption pair.

Comparison:

Searching for audio recall score:
[0.07421875 0.24902344 0.36621094]
Searching for image recall score:
[0.06933594 0.22363281 0.31933594]

Baseline

Searching for audio recall score:
[0.09375 0.26757812 0.36328125]
Searching for image recall score:
[0.08984375 0.22949219 0.35449219]

Result Using New Loss Function

Modification of Model Architecture

We proposed to augment the spectrogram feature vector by explicitly modelling temporal information in another branch parallel with the baseline architecture. Suppose our spectrogram is an image of size (num_freq, num_time, num_channel). First, three sets of vertical frequency filters are learned to collapse the frequency axis. Then another set of filters are learned for local temporal information. Horizontal max pooling is then applied, before the output is fed into the next layer, which is called "Temporal Convolution Networks".

This kind of network is basically a 1D fully convolutional network with three modifications, causal convolution, dilation and residual block. Causal convolution is used to model the relationship $f(y) = f(1..y-k)$, while dilation is used to exponentially increase the receptive field. Residual block is used between previous layer and current layer to learn modifications to the identity mapping, rather than full transformation.

An illustration of a single TCN stack is shown below.

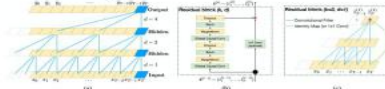


Figure 1: Architectural elements in a TCN. (a) A dilated causal convolution with dilation factors of 1, 2, 4 and filter size 3-3-3. The receptive field is able to cover all values from the input sequence. (b) TCN residual block. An L1 convolution is added when residual input and output have different dimensions. (c) An example of residual connection in a TCN. The blue lines are when is the residual function, and the green lines are identity mapping.

We used two stacks of TCN, which, combined, gives a total receptive field of over the whole temporal dimension, so the last temporal vector of the second TCN stack is used as our temporal feature vector. This temporal vector is concatenated with the caption-wised pooled vector of the baseline architecture, with the number of feature maps reduced. The concatenated feature vector is then fed through a dense layer, in the hope to learn a meaningful correlation among the concatenated vector.

Results

There are two ways to use our developed machine learning model.

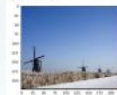
- 1) The input query is an image, and the output is related captions.
- 2) The input query is a caption, and the output is related images.

For better understanding and visualization, we transform audio captions into ASR text captions and also provide ground truth images with returned text captions in pairs here. And an example of search result is below. Shown on the top is an image with its ground truth audio transformed into text caption form. Below are its five highest scoring audio files transformed into text caption form.

Input Image:



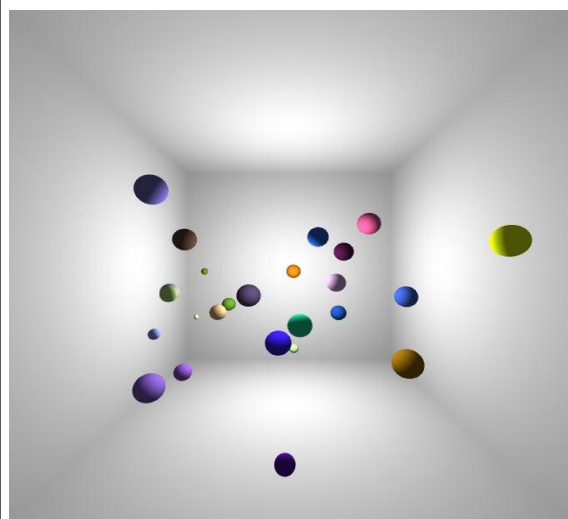
Search Results:





Interactive Computer Graphics Projects collection

- Created a 2-D animation of a dancing Illinois Victory Badge.
- Built a flight simulator game. Generated the terrain using the Diamond-Square algorithm. Used Blinn-Phong illumination model and Phong shading with a colormap. Added a weather-change feature.
- Wrote an app that loads the Utah teapot and renders it with environment mapping. Added the Quaternions option for the player to modify the teapot position.
- Rendered a system of rainbow-colored bouncing spheres in 3D with the effect of gravity and friction.



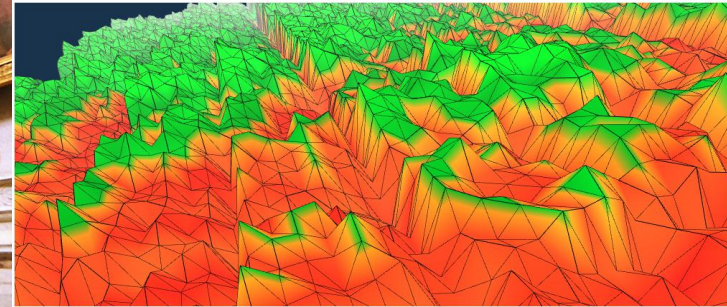
ZXY Flight Simulator

Controls:

Up/Down arrow keys: pitch; Left/Right arrow keys: roll

A: yaw left, D: yaw right

+: speed up, -: speed down. Z: fog on, X: fog off



Rendering Parameters

Wireframe Polygon Polygon with Edges

Collaborative Filtering Based Course Workload Prediction system



$$p_{a,i} = \bar{r}_a + \frac{\sum_{u \in K} (r_{u,i} - \bar{r}_u) \times w_{a,u}}{\sum_{u \in K} w_{a,u}}$$

	cs125	cs233	cs241	cs374 ...
student1	6 (hrs/week)	10	25	16
student2	8	12	N/A	20
...

A terminal-based application leveraging user-based collaborative filtering algorithm to help students wisely plan their semester course schedule based on their expected course workloads.

Type in your email
for example, 'tracy@gmail.com'
>>yms34@illinois.edu

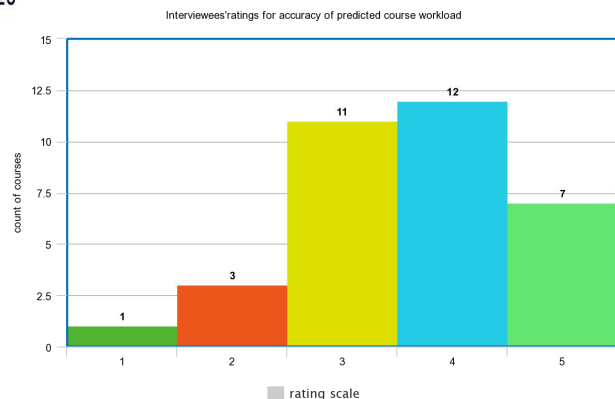
User	Predicted courses	rating for each course
1	CS225, CS374, PHYS211	4, 4, 3
2	GEOG101, ECON103, CS125, STAT400, NPRES100	3,3,2,4,2
3	PHYS212, PHYS213	5,4
4	CS461, CS357, CS374	3,5,5
5	CS425, CS421, CS498WEB, HORT107	4,3,4,1
6	CHLH101, CS225, CS233, CS241	2,5,3,3
7	ECE220, ECE385	4,4
8	MATH286, PHYS213, CS473, CHEM102, PHYS214, CS446	3,4,4,3,5,3
9	CS410, CS411	4,5
10	CS241, CS126, CS427	3,5,4

Type in your taken courses with corresponding workload separated by spaces
for example, 'cs125:10 cs233:11 cs225:16 cs241:25'
>>cs125:10 cs233:8 cs225:20 cs241:25
(51, 220)

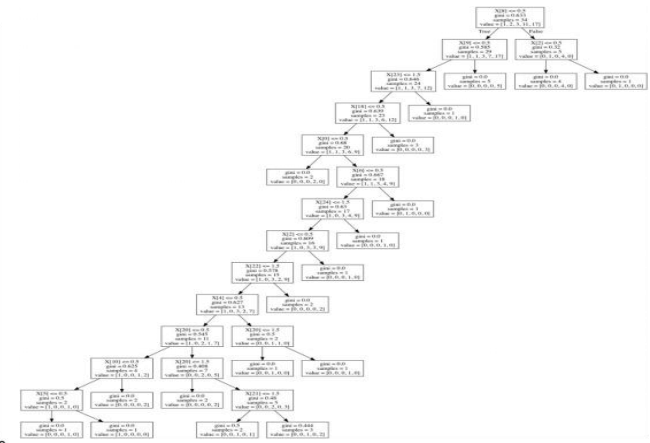
Type in course you plan to take
for example, 'cs242 cs210'
>>cs374
(yms34@illinois.edu', 50)
(cs374', 11)
(15.75, 11.507705316731125, 1.475216166015602)
prediction workload is 23.550690896583774hours

```
true_val = [12, 6, 11, 3, 10, 6, 4, 8, 1, 18, 1, 3, 20, 16, 5, 8, 19, 2, 12, 9]
pred_val = [12, 12, 7.65, 4.57, 9, 6.6, 7.1, 6.5, 3, 4, 20.1, 2, 1, 6.2, 20.5, 13, 5.5, 8.96, 23.2, 10.5, 8, 9.5]
rms = sqrt(mean_squared_error(true_val, pred_val))
print(rms)
```

3.170875273485225



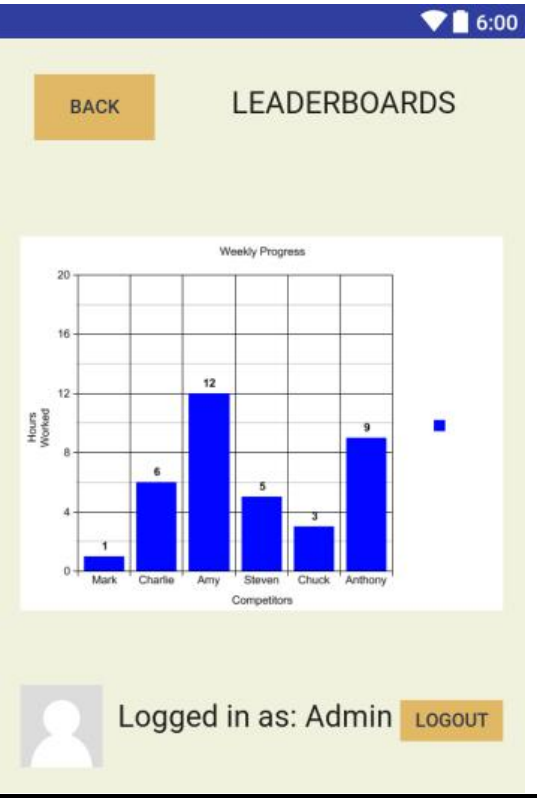
-
- The figure is a scatter plot titled "Datasets" with axes labeled x_1 and x_2 . The x_1 axis ranges from 0 to 6, and the x_2 axis ranges from 0 to 6. There are two distinct clusters of data points. The first cluster, on the left, is centered at $(2, 2)$ and is marked with a red star and a red circle. It contains five data points at $(1, 2)$, $(3, 2)$, $(2, 1)$, $(2, 3)$, and $(1, 3)$. The second cluster, on the right, is centered at $(5, 4)$ and is marked with a blue star and a blue circle. It contains five data points at $(4, 4)$, $(6, 4)$, $(5, 3)$, $(5, 5)$, and $(4, 5)$.





— Productivity Enhancer mobile app

An Android mobile application that enhances individual studying/working efficiency.



—

GITAPP

A mobile app that allows users to visualize their personal GitHub information, such as their repositories, followers, etc. Implemented in Java.

