Spell Check

Denis, Neil, Katarzyna, Anthony, Brandon

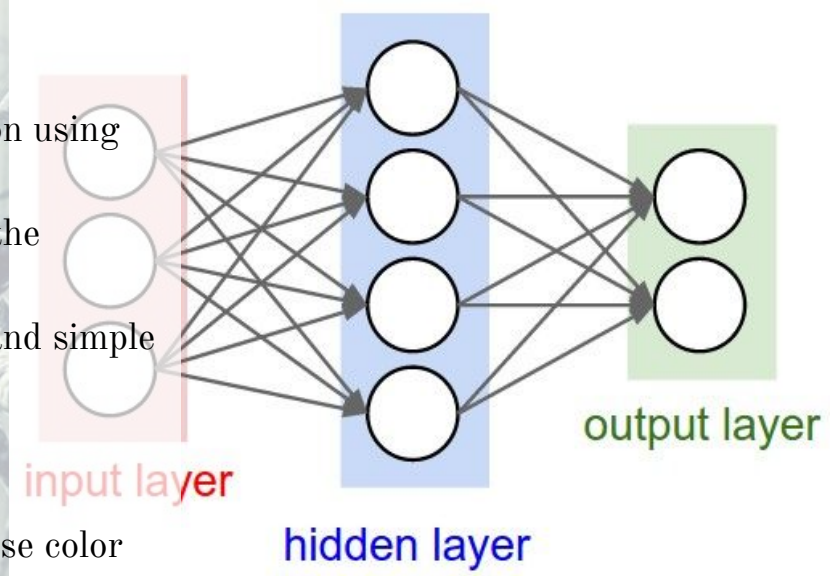
Introduction

- Quick recap:
  - Goal: check spelling of real-life text
  - Last time: talked about various approaches (machine learning, greedy algorithms)
- This time:
  - Focused our research effort entirely on machine learning (specifically, neural networks)
    - Discarded Greedy Point Match algorithm

Imagine that.

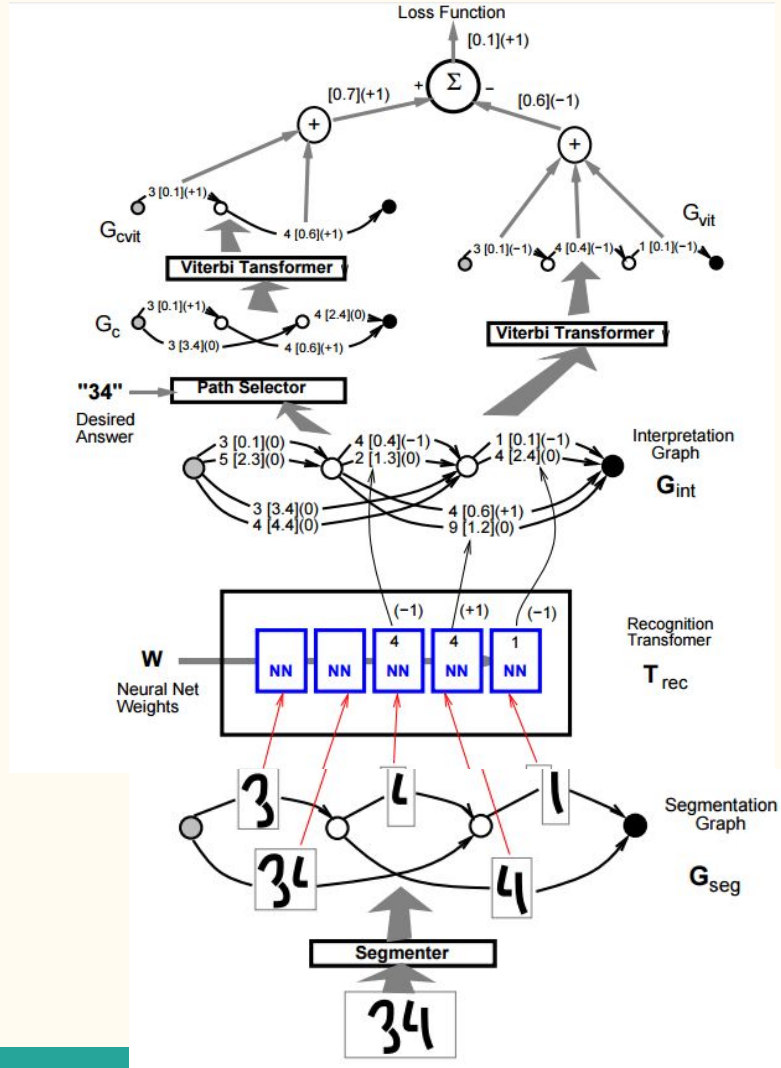
Introduction

- Brainstorming & Learning:
  - Learned how neural networks worked
  - Zoned in on a paper on handwritten text recognition using neural networks
  - Started a cross-platform C# project to implement the algorithm
  - Implemented the neural network in C# to understand simple written text
  - Tested the code using static images
    - Trained and tested with static images
    - Next up: train with UJI Pen (1364 chars), use color images (preprocess + segment), move to real-time, port to HoloLens (C# + Unity)



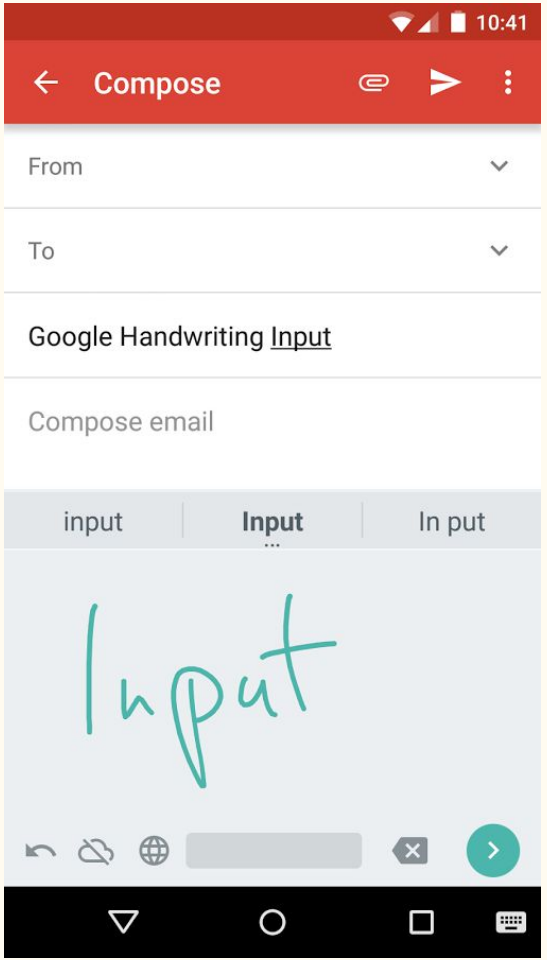
Prior Works

- LeCun et al. 1998
  - Field Location Transformer: heuristically extract rectangular zones
  - Segmentation Transformer: cut zone into all possible groupings of ink
  - Recognition Transformer: a convolutional NN to classify segments in graph into ASCII chars
  - Composition Transformer: takes recognition graph and a grammar graph to filter for valid sequence paths
  - Viterbi transformer: select path with lowest penalty, aka the best grammatically correct interpretation.



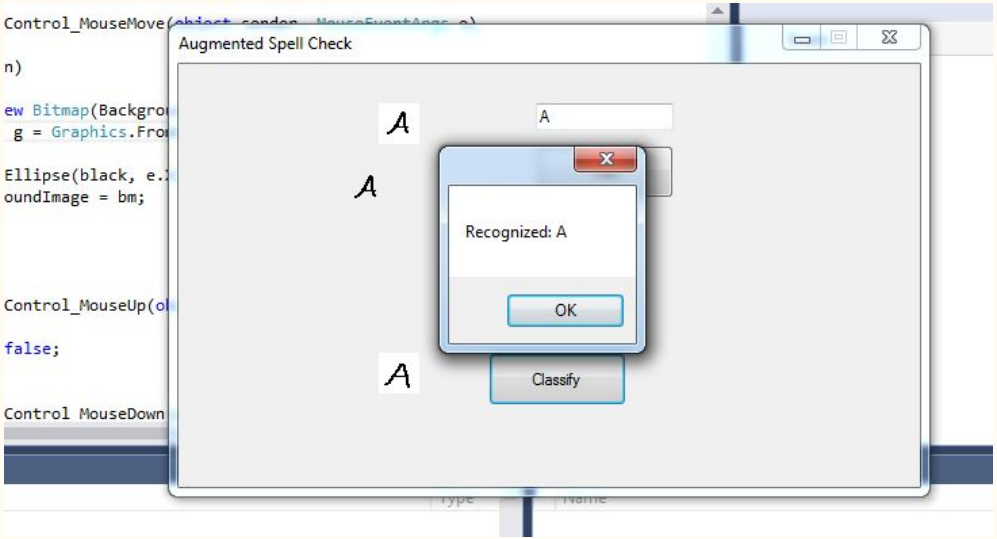
Prior Works

- LeCun et al. 1998 (from prev. slide) tested their algorithm using various NNs, CNNs, etc. -- 1-3 layers, various hidden units, elastic distortion, etc.
- Keysers et al. 2016 is an overview of Google's handwriting recognition system that supports 22 scripts and 97 languages.
  - Involves dozens of steps to be setup and trained.
  - Extremely fast, supporting variations for both desktop, online, and mobile use.
  - Novel aspects: unified time and position-based input interpretation, trainable segmentation, etc.



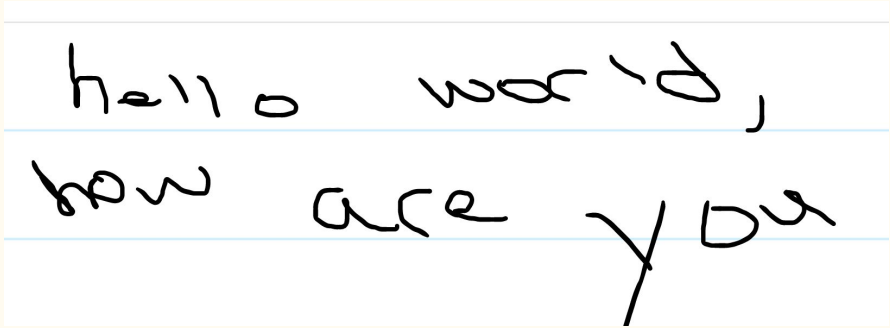
Experiments

- Created a 900-900-8 NN.
- Built a MS Paint-like application user control (with support for only 1 color).
  - It is 30px by 30px (900 inputs). Larger images significantly increase the number of input nodes, and this isn't a CNN
- Draw a char, hit Train. Pixels are converted to array of 0's and 1's to train NN with.
- Hit Classify to identify possible char from grammar



Myscript - Commercial

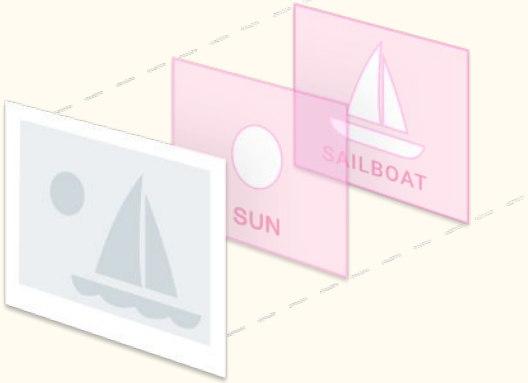
- MyScript is editing software.
  - Provide online handwriting recognition-based applications and SDK's.
- Converts digital ink into text format
- Solves one of our main concerns of handwriting recognition
  - Can't use digital ink for our project - Drawback



hello world, how are you

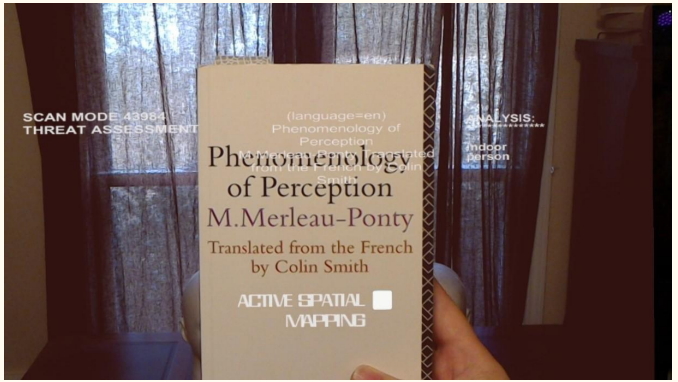
Word Lens - Commericial

- Augmented reality translation application
- Recognize words after taking a picture and allow user to translate them
- Bought by Google in 2014
- Can be recreated using
  - Google Cloud Translation API
  - Cloud Vision API
- Incorporate Cloud Vision API into our project
  - Don’t need the rest of the functionality



Prior Works

- “Building the Terminator Vision HUD in HoloLens”
  - By Windows Apps Team
  - Essentially a blog post detailing how the team went about creating a HUD from the *Terminator* movie utilizing HoloLens. They also went through the effort to include an OCR aspect to it.
  - The blog also serves as a pseudo-tutorial to developing within Unity



- https://blogs.windows.com/buildingapps/2017/03/06/building-terminator-vision-hud-hololens/#0yotWfZLQm8OFsf5.97

What does Spell Check do?

- Well..... It checks your spelling.
  - Similar to Microsoft Word
- Parse and analyze handwritten text
  - Uses a NN to detect handwritten text

Benefits of Spell Check

- Spell check's benefit is really simple.

  It's here to check your spelling.

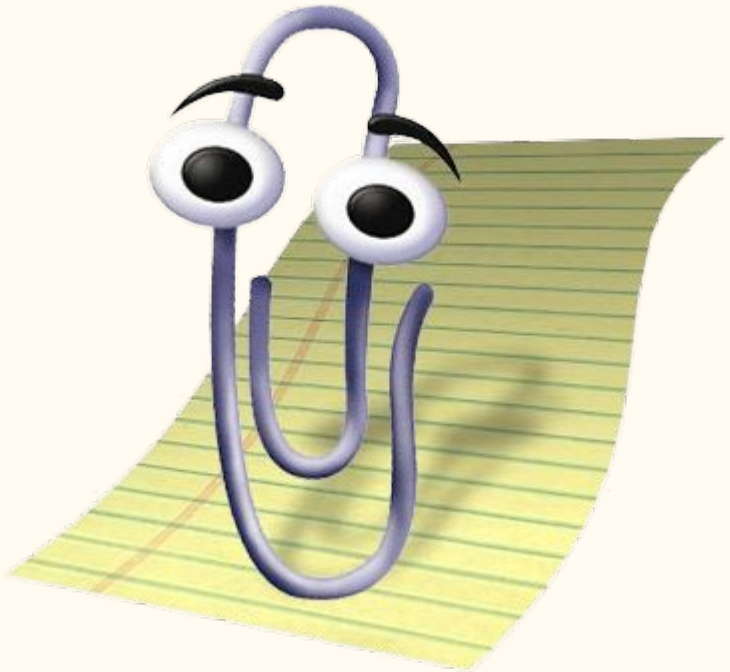
- Ex) Writing notes in class

Target Market

- Three main target groups
  - School children
  - People with writing disabilities
  - Foreigners
- Want to make it easy enough for children to use
- Display user friendly, so someone doesn't need english fluency for use

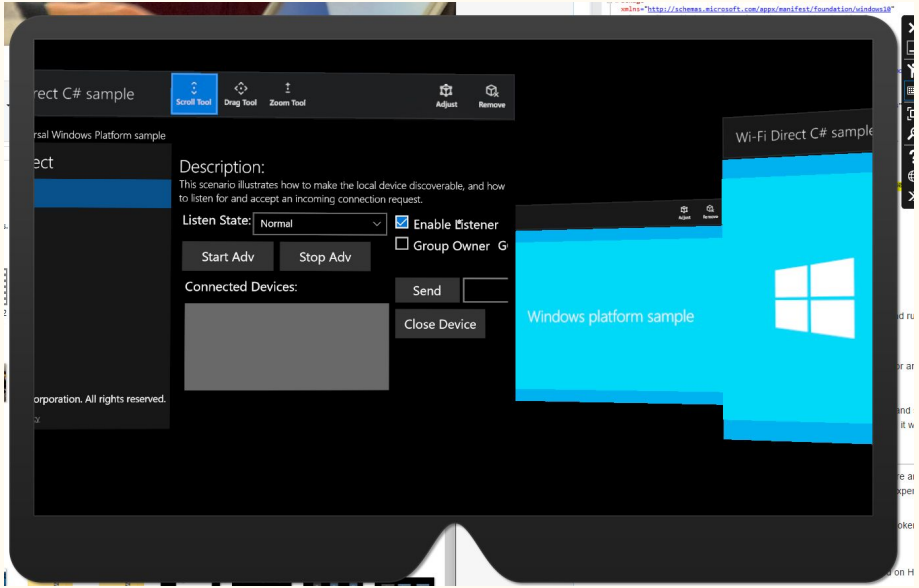
Why is this useful?

- Helps out the target audiences mentioned previously.
- Helps everyday people from easy spelling mistakes.



Problems we have faced

- Handwriting
  - Being able to recognize different types of handwritten text
- Spell checker behave correctly
  - Identify the right words for correction
- Being able to test our project
  - Emulator is bulky and takes a long time to install and run.



Problems we *might* face

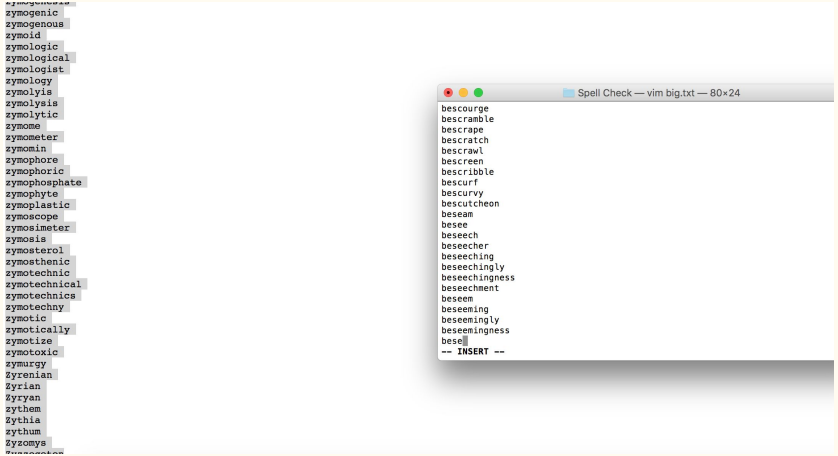
- Underlining misspelled words with a red line
  - Being able to place a line under the recognized misspelled word in HoloLens
- Display proper spelling
  - For each misspelled word having it recognize the correct spelling that is closest to the underlined word by giving proper suggestions
- Coordinating our efforts
  - Setting up proper meeting times
  - Making sure everyone has done the part they are responsible for

Simple C# Spell Checker

```
speling => spewing
korrecter => corrector
korrect => correct
acess => access
supposidly => supposedly
correct => correct
```

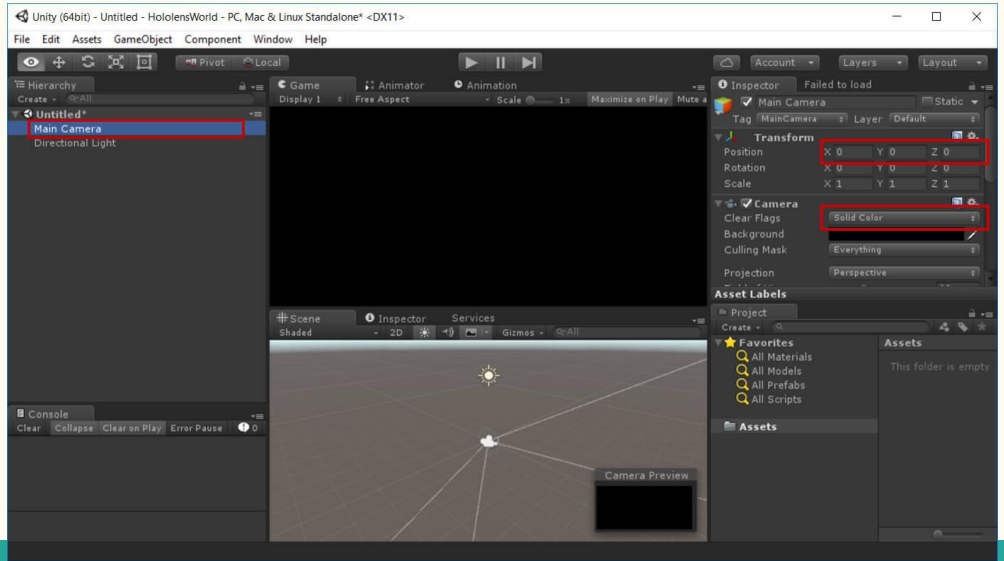
- I created a simple dictionary based spell-checker
  - Uses large bank of words checks against them
  - Creates list of candidates and narrows down list
- Currently works on a limited level
  - Obvious bugs with selection
- Expand and build a more robust system
  - Correct for certain behaviors

```
2 references
public class Spelling
{
    8 references
    private Dictionary<String, int> _dictionary = new Dictionary<String, int>();
    1 reference
    private static Regex _wordRegex = new Regex("[a-z]+", RegexOptions.Compiled);
```



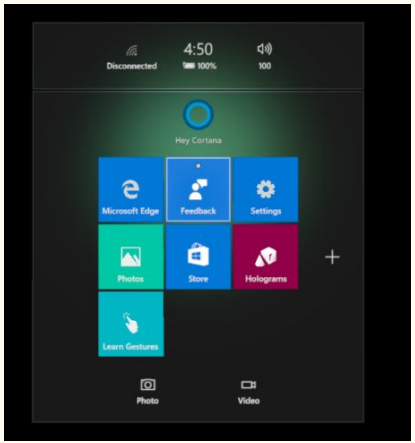
Unity 3D work

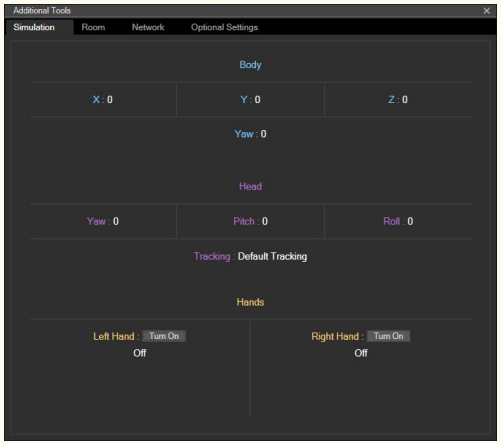
- Development for Spell Check will have to be done through the Unity 3D engine (v5.4+)
- We've installed the proper Unity version and have done a little demo work within it
- We've made the "squiggle" appear



HoloLens Emulator

- Emulator is run through Visual Studio (2015 v3)
  - Will allow us to test the app without the actual HoloLens in addition to more precise debugging
  - The emulator has the ability to move the user's head/display in order to mimic real movement





Work done so far

```
use Test::Spelling;
add_stopwords(<DATA>);
all_pod_files_spelling_ok();
```

- Automated testing scripts
- Testing scripts for the visual character recognition
- Feeding a set of image files and output files to test if the actual output matches input
    - Each character is examine separately and and checked for correctness
- Testing spell checker
- Feeding a set of misspelled words and testing to see if it identifies the misspelling and identifies the correct word

```
while (<$data>) {
  chop;
  my ($mis, $cor) = split /\t/;
  die ":$mis:  :$cor:" if $mis !~ /^[a-zA-Z\']+ ?[a-zA-Z\']+$/;
  die ":$mis:  :$cor:" if $cor !~ /^[a-zA-Z\']+ ?[a-zA-Z\']+$/;
  print $out "$cor\n";
  my $res = <$in>;
  chop $res;
  if ($res) {
    print R "$mis\t$cor\t-1\t-1\n";
    $res = <$in>;
  } else {
    print $out "$mis\n";
    $res = <$in>;
    chop $res;
    if (!$res) {
      print R "$mis\t$cor\t-1\t-1\n";
    } else {
```

Timeline:

	Neil	Anthony	Denis	Brandon	Katarzyna
Week 11	CNN	Unity 3D	Finish Simple SC	Help with paper	Testing Script SC
Week 12	Segments	Help with Paper	Finish Complex SC	Unity 3D	Testing Script CNN
Week 13	Video stream	Finish up Unity 3D	Help with paper / Complex SC	Finish up Unity 3D	Testing Script CNN
Week 14	HoloLens	Testing	Meet with group and Combine	Test and Debug	Help with paper
Week 15	HoloLens	Testing	Testing	Test and Debug	Help with debug

Any Questions?