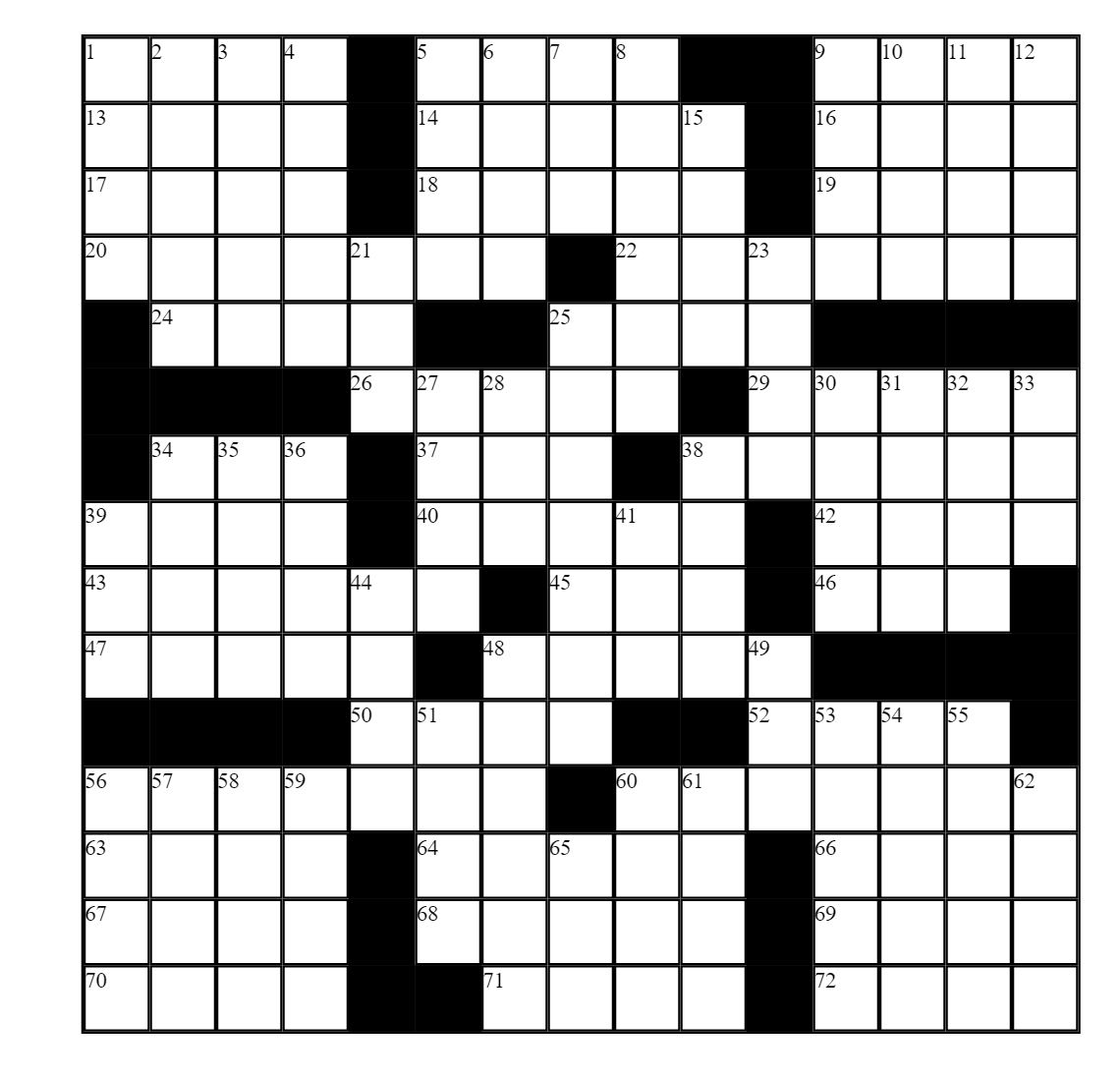
Ryan Helmlinger

Sys-Lab Journal

September 16th, 2016:

So far I have made a user interface website that allows you to create a crossword puzzle. It allows you to add the black squares wherever they appear in the crossword that you are trying to replicate. It also fills in the symmetric spot for the black square, since all crossword puzzles are symmetrical. Then it automatically fills in the assigned numbers to fill out the board like this:



This website also lists all of the across and down spaces created and has an input field to add hints. This changes automatically when a new square is added.

My next step was going to be to use an online API to solve the crossword puzzle hints. However, I haven’t found any online APIs. The closest thing I have found so far is a brief journal about how to use the Google Custom Search API. Another idea was maybe to use wolfram alpha, but in order to test if that works I need to buy the pro version of wolfram alpha.

September 23rd, 2016:

I have still been looking for a good online api to use, but haven’t found any new methods besides what I listed in the previous week. I also started working on getting my website to communicate with a python file.

September 30th, 2016:

The python file wasn’t working well by basic communication with the JavaScript. I did some research online and found out that my python file needs to create a web server in order to communicate with a website. In order to do this, I need to use the python cgi api.

October 14th, 2016:

I’ve still been researching and understanding the python cgi api. I’ve also been trying to think about how to fit the words into the crossword puzzle in the most efficient way. I could use brute force, but it would more than likely be too slow, so I’ve been trying to think of better alternatives.

October 21st, 2016:

I had originally been using 15x15 crossword puzzles for my code, but I found out that 21x21 crossword puzzles were also very common. So now I’ve been working on making the crossword puzzle for any size that’s inputted. My code was previously hardcoded, so I had to change much of it in order to make it work for any size. I assume that this will also make the Computer Vision part trickier since it won’t necessarily know the size (unless the user enters it first).

October 28th, 2016:

I’ve been trying to connect the web server to a python script, but the python cgi api keeps returning errors. I’ve spent a majority of the week trying to get this part to work. The file seems to be communicating with the server, there just seems to be a couple lingering errors left that I need to fix.

November 4th, 2016:

I started looking more into solving the hints. The only possible online API might be Wolfram Alpha. I also read a paper by David E. Goldschmidt and Mukkai Krishnamoorthy, which further explained how to use the google custom search api. This might be the path I have to take in order to try to solve the hints. However, this method might not work 100% of the time.

November 11th, 2016:

I wrote a program to look at possible word choices and fit them in a crossword puzzle. With only correct choices, it always works. However, when I add false options, it only works around 90% of the time. I need to fix the recursive bug in order to get it to work all of the time.

November 18th, 2016:

I finally determined that there are no online APIs that directly solve crossword puzzle hints. So now I’ve started researching out to use the Google Custom Search API to do it.

December 2nd, 2016:

I started using the Google Custom Search API. I got the search engine to work, but now I’m not quite sure how to actually get the words from it. Right now it only gives me the links to the websites with the words, it doesn’t actually read them out.

December 9th, 2016:

I found an online python API Dictionary that find synonyms for words and phrases as well as definitions. I got it to work, but the program only could work with words, it didn’t understand any phrases, which are mostly what crossword puzzles are.

December 16th, 2016:

I started looking at the Google Words API and it works with some phrases. However, I will likely have to parse the clue in order to get decent results. This method seems to be the one that works the best so far.

December 16th, 2016:

I started looking at the Google Words API and it works with some phrases. However, I will likely have to parse the clue in order to get decent results. This method seems to be the one that works the best so far.

January 6th, 2017:

I am currently having a hard time making progress on solving the clues, so to give it a break, I have started looking into the Computer Vision portion. I plan to use OpenCV2. I installed it and got it working properly.

January 17th, 2017:

I become a lot more familiar with OpenCV2. I can do several things with it, but I am only now starting to shift my focus towards using it to read a crossword puzzle. This part will be tricky, but I hope I will be able to get it to at least recognize the grid.

January 20th, 2017:

I’ve been reading several articles on language processing for crossword puzzles. I’ve gotten some ideas of where to start, but I am still currently wrapping my head around the semantics part. The Google Words API still seems like my best option so far.

January 27th, 2017:

I’ve read in several places that scraping the web for previously used clues and answers works better than many alternatives. Thus, I’ve started reading up on how to create my own program to scrape the web in order to find these. I haven’t used anything like this, so it may take time to get it properly working.

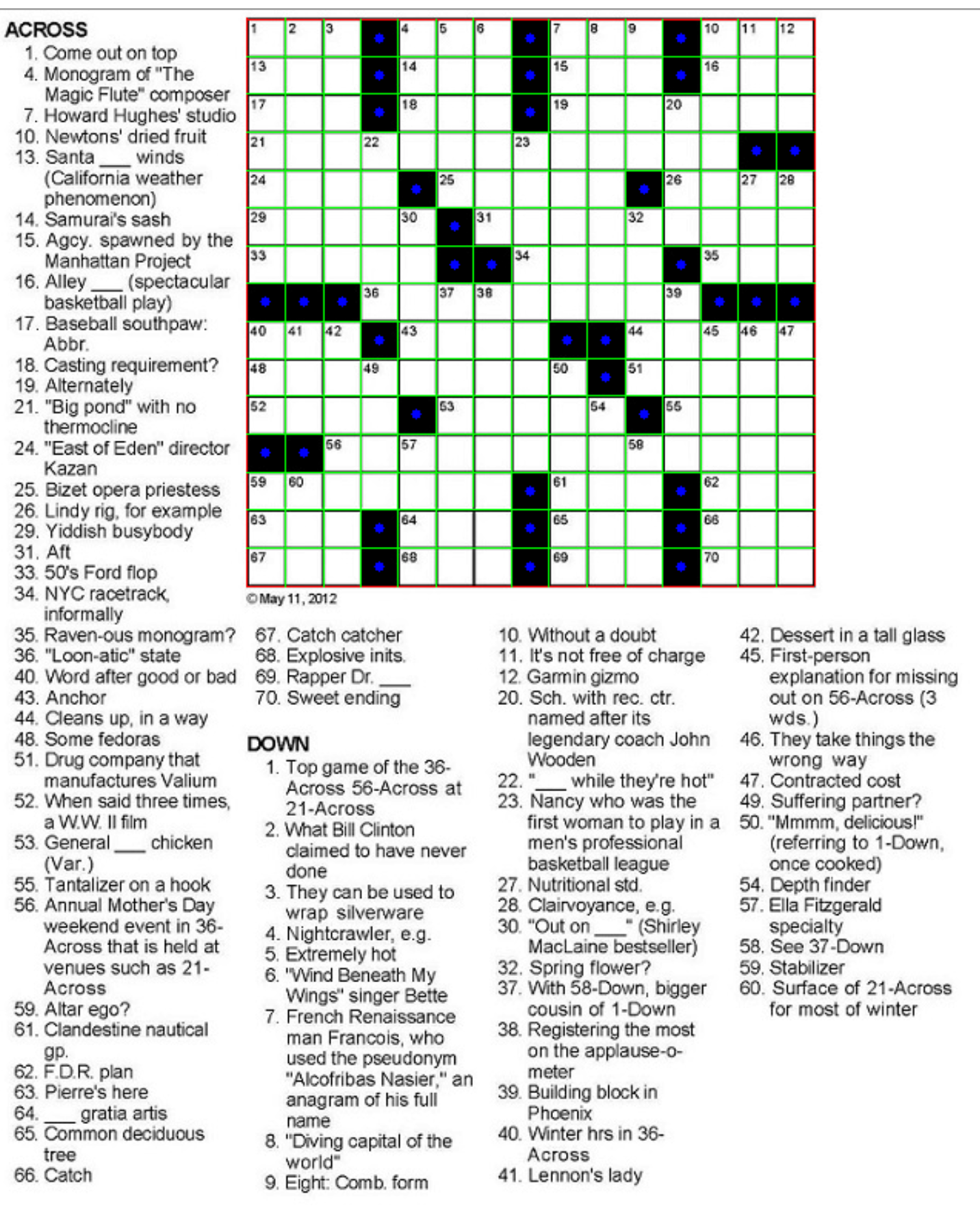
February 3rd, 2017:

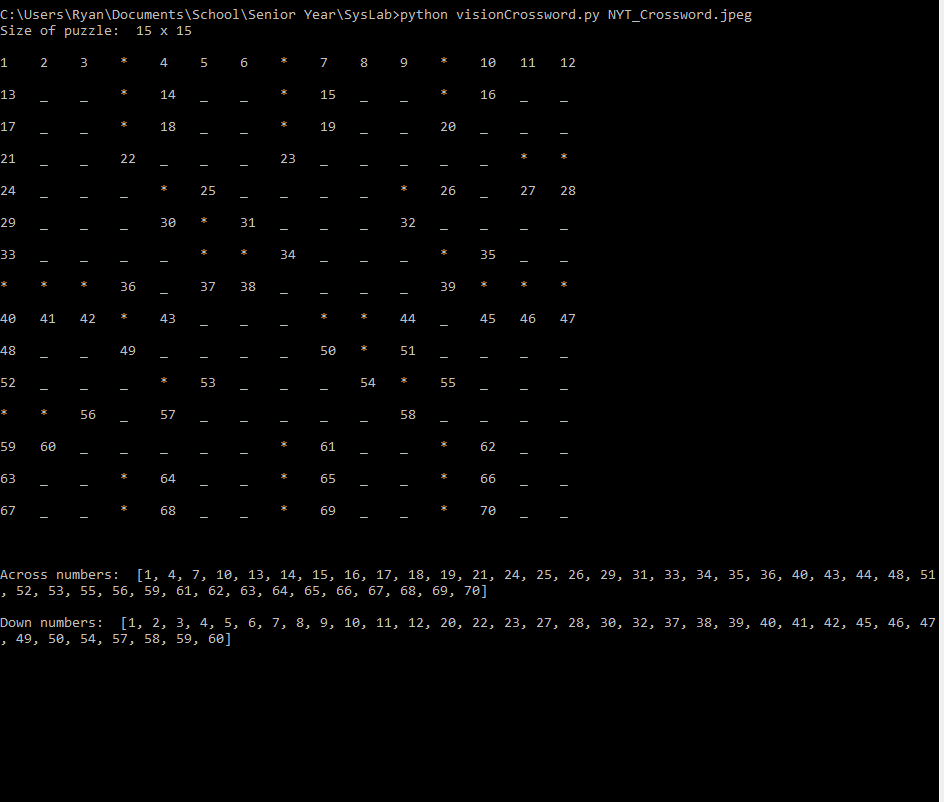
I wrote a python script that scraped data from Dictionary.com. The data is based on clues from the New York Times. It currently correctly gets the hint about 90% of the time. After this, I started working on the computer vision component. The picture below shows that the program was able to find the outline of the grid. It highlighted it in red.



February 10th, 2017:

Now my computer vision program finds of the lines inside of the crossword puzzle (green lines). This allows the program to determine the length and width of it. It also determines whethere a square is empty or not (blue dots), and then the program automatically numbers the squares itself due to the black squares. Below are the pictures of the outputted image, and the terminal output of understanding the image.





February 17th, 2017:

I’ve tried using several programs like OCR and Pytesseract, but I’ve had problems installing the programs. I also did what u suggested and erased the grid so that it would be easier to read the text. However, I still need to install the programs correctly first.

February 24th, 2017:

I’ve still been working on installing OCR and Pytesseract. Plus, I spent part of the week working on the presentation that I made on earlier today.

March 3rd, 2017:

I worked on trying to figure out special cases in crossword puzzles. The special cases include underlines, questions marks, puns, quotes, and clues that reference other clues.

March 10th, 2017:

Still worked on figuring out the special cases to the crossword puzzles. One example is from 60-Down on the crossword pictured above. The clue is “Surface of 21-across for most of winter”. The unique punctuation and puns, however, exist more frequently in the puzzles so I will be focused on them more.

March 17th, 2017:

I finally got OCR and Pytesseract installed properly. I can compile and run the base code without any issues. And it works with a plain picture of text. However, it doesn’t work with the crossword puzzle image and I’m trying to figure out why.

March 24th, 2017:

I tried formatting the picture of the crossword puzzle in many ways to try to get it to work without error. However, it only seemed to work when I took small snippets of plain text. When I looked at a couple clues in a row, it seemed to error because it had trouble reading it, not because of formatting. I need to figure out a way to get past this. Or I’ll just have to have the user select each clue t read in.

March 31st, 2017:

This week I had a setback. My computer crashed and wouldn’t start windows properly. Lucky I had my files backed up, but I still couldn’t start the computer. As a result, I lost everything I had installed, including python, openCV2, Pytesseract, OCR, and many other programs necessary for this research project.

April 7th, 2017:

This week I have been working to re-install everything that I lost last week.

April 28th, 2017:

The past few weeks I got the image reading to mostly work and created the user interface. The user interface works and is very intuitive. A picture of it is below. However, you can also see in the picture a few of the clues aren’t completely correct. For example, 10 across is “Without a doubt” but is read in as “Vlnthout a doubt”. In order for the clue reading to work as planned this final part needs to be fixed for the puzzle to be completely solvable.

