Test conditions

127 captchas tested. 2 models used. Google speech recognizer and Assembly AI. An error was recorded if the model missed a character in the captcha or if the model gave a character not in the captcha. Some captchas could not be recognized at all. These were recorded as fail below and were scored as 4 errors. A score of 0 indicates no errors were made and the model solved the captcha correctly. The scores shown are the best scores for the model, meaning the output of the model was corrected as much as possible using the auto-correct dictionary before the score was recorded.

Test results below

CAPTCHA	Google SR	Assembly AI
4wctx	fail/4	0
4wkk	3	2
5p3wd8	fail/4	0
6ap8bh	0	1
6d8n	3	1
6umtm	4	1
8d8e	1	1
8d8h	2	1
8krcx	2	1
8n3atr	5	1
8rve3	4	4
8uh8j6	3	1
8vk8x	5	0
8wwe	2	1
9hat	3	3
9hcsst	fail/4	4
9j4tak	5	1
9nr58	4	1
9vuw	4	4
33cwk	0	1
64usy8	5	0
66x9vm	4	1
88jhc	4	4
89d4	1	3
459jpa	fail/4	1
599ty	0	1
a9h6	fail/4	2
amep	3	1

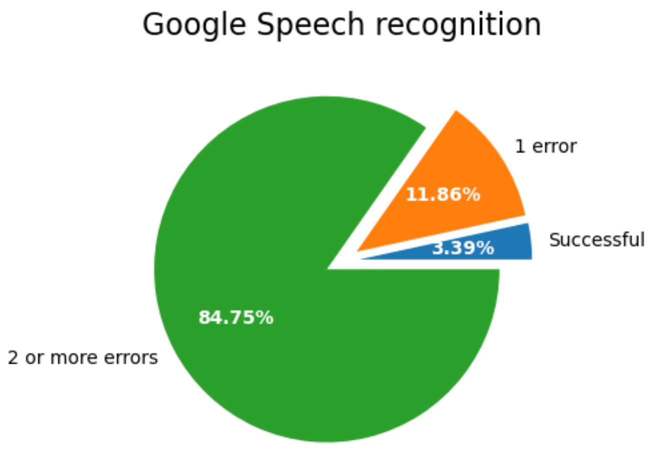
b9m99	3	0
bep9sd	3	1
bmp9	fail/4	3
bnh6pa	3	1
buct4v	3	2
bv8p8j	3	1
bveu	fail/4	1
by8h	1	2
c8ekcn	fail/4	2
c9b5r3	4	3
ck99u	3	1
ckkdw	3	2
cnc4	1	0
cr5mu	2	1
crbv3h	3	fail/4
ct46y4	2	0
cu48y	3	1
cyn8p	3	2
d3e34	3	0
d83h	3	3
dntwru	1	1
dp8n4	4	1
dw4sc8	3	1
dysp	3	1
e8hj	2	2
e43n	1	1
ea8d	3	2
edtxn	fail/4	1
eebr5	5	0
es6a	3	1
ewhab3	1	0
h4d8	2	3
h33a3d	3	1
hb5ew	3	1
hbhcr	3	1
j68b	1	1
jkm5rn	2	1
jnm6np	3	0
jt5ctc	3	2
ju3v	2	1
jutd6	fail/4	0

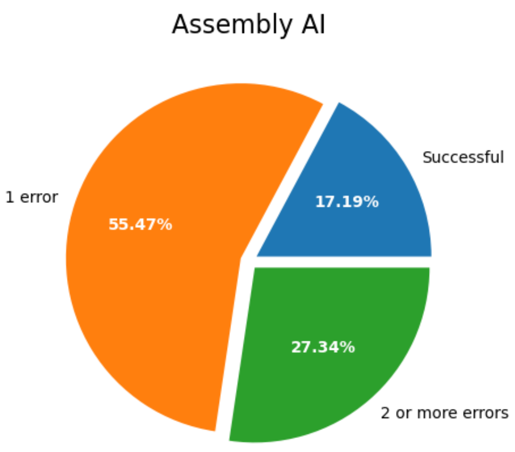
jwdc4	1	0
k8ayny	2	1
kvyp	3	1
mhnxe	fail/4	1
mm49m	3	1
mn5r	3	1
mnhvx	5	1
mu396s	3	1
mynjt	3	1
n4s9	3	0
n864	2	1
nakdwp	fail/4	1
nk3m	2	1
nyku5c	2	1
pkpya9	4	0
pu9auy	6	1
r9eb9	fail/4	0
rb6y8e	1	1
rdhse5	4	2
rdmcck	fail/4	1
rnhkbb	fail/4	1
rnvhh3	fail/4	1
rybrw	fail/4	1
scuupr	2	1
sn3j	2	2
t4ucde	3	2
t6uj	3	1
t9ar	4	1
tb6p	2	2
tbr8e	4	2
tp63	3	0
ttsxhbv	3	1
ty6s	1	1
u4dh	fail/4	fail/4
u8xhbx	fail/4	2
u66cdm	fail/4	3
ubpt	fail/4	1
uj8c	1	1
umhk	0	1
uxd9	3	0
v4p96t	fail/4	2

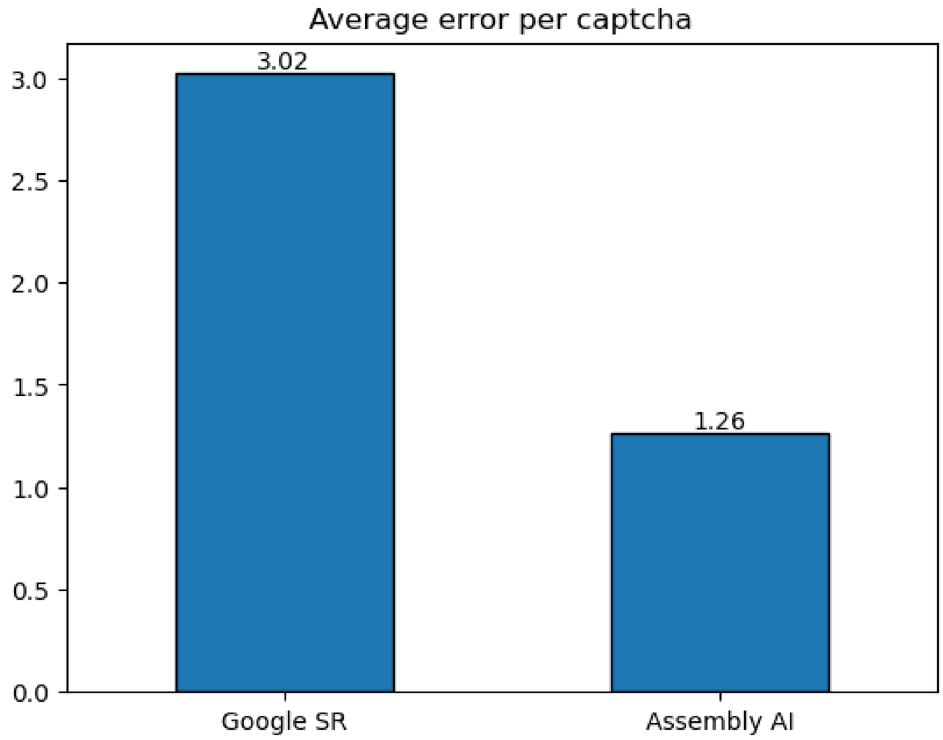
vcedc	3	1
vry3v6	4	0
w64xb	4	1
w964v	2	1
wdehu	4	2
wr5pd	4	0
wwwe8e	fail/4	fail/4
wy9r	fail/4	1
x9cc	1	2
xanm	fail/4	1
xtu6vd	3	1
ydnk	3	1
yewrp	fail/4	1
yj5b58	3	0
yj6nvb	4	0
ymyymv	4	2
ytru	1	1
Total Errors	383	160

Performance Metrics

Below: The respective percentage of captchas each model solved successfully, made one error while processing and made two or more errors while processing.







Factors affecting accuracy

The phonetics of some alpha numeric characters are similar to others, which can cause difficulties distinguishing and correcting them especially if they appear together in the captcha. For example, “a”, “8”, “h”, “hey” were sometimes misinterpreted interchangeably as well as “6” and “x”.

The characters are sometimes spoken in an uneven/staggered rhythm. In some instances, the recognizer would miss only one of the characters because there was no break between the spoken characters. (For example, see captcha 89d4 using google sr and its associated audio signal for reference). On some occasions, both models, especially assembly ai, would miss only the last character of the captcha. This was likely because the recording ended abruptly without giving the model enough time to interpret the final character. A possible solution, that could improve accuracy is to use the librosa time_stretch() function or another audio software program to stretch the file out and pass the resulting file to the recognizer to give it more time to catch the last character.

Captchas sometimes had a reverb/strobing effect applied to them (Example captcha bmp9). This made them harder for google sr to solve.

Implemented solutions

Writing a dictionary ad hoc as the tests were run corrected some misinterpretations and improved the accuracy and output of the model.

Some of the captchas ended abruptly just as the last character was read and the model often missed the last character. An attempt to fix this was made but was not successful. By adding 2 seconds of silence to the end of the audio file, it was hoped the model would have more time to catch the final character. However adding 2 seconds of dead air simply appears as a flat line when plotted visually, and the model will recognize nothing from this. This method, in order to work, would require the original recording to run for a few more seconds before ending.

Some of the recordings had noise in the background. A line of code could be added to adjust the speech recognizer for ambient noise; however, this was generally found to be counterproductive. It often caused an unsuccessful read or cut out one of the characters in the captcha which would otherwise have been recognized.

Finally, the threshold of google sr can be set using:

```
r.energy_threshod =
```

Meaning the model will listen and attempt to interpret all sound above this level only. However, this didn’t seem to improve accuracy.

Suggested next steps

- Investigate other speech recognition models such as ibm watson, deep speech and whisper open ai. Run tests using these models. Some models are not open sourced and may require a cloud computing account to obtain an API key.

- Trial combining two models together. I.E Combine the output of both models and print common characters once only. A common problem found during the tests would be one model would miss the first character whilst the other model would miss the last character. (see test results were each model made one error each)

- Integrate final program into a toolbar extension