welcome to **geogenics**

introduction

‘welcome to geogenics’ is a game where you work for a morally dubious company which specialises in ‘artificial biodiversity’: producing new animals to replace species lost to climate change. Your job as a ‘turing tester’ is to do quality assurance by trying to determine which animal names are machine generated and which ones are real, existing birds.

This project grew from me somewhat accidentally stumbling upon a list of names of every known bird species. Inspired by projects using machine learning to generate new words such as paint colours[i] and British place names[ii], I decided to try to use machine learning to generate some all-new bird names. I quickly set up a project on Google Colaboratory and ran my list of bird names through it. The results were fantastic. The program was spitting out names like ‘Pohnpei Lorikeet’, ‘Fork-tailed Drongo-Cuckoo’ and ‘Snethlage's Tody-Tyrant’.

However, upon closer inspection, it turned out that the algorithm had learned my bird list perfectly. The names it was spitting out were real birds. Turns out, reality is stranger than machine generated dreams!

After some tweaking of parameters, I was able to generate new, unique bird names. But I was intrigued by my inability to distinguish between the real and the fake. I also wanted to explore these fake birds further. How did they come about? Who or what created them? What does a world look like where real and fake wildlife coexist? How can one determine ‘fakeness’ anyway?

This project started out as fun and light-hearted experimentation with machine learning, but the more I worked with the datasets, the more I was also struck by the sheer number of bird species out there, and how incredibly diverse and weird wildlife on Earth is. At the same time, we are losing species at an alarming rate. It might be a fun game, but how could it not be political?

I wanted to use this small platform to draw attention to climate justice. Hopefully, some people playing the game will be inspired like I was, and I want to encourage taking action against climate disaster.

access and accessibility

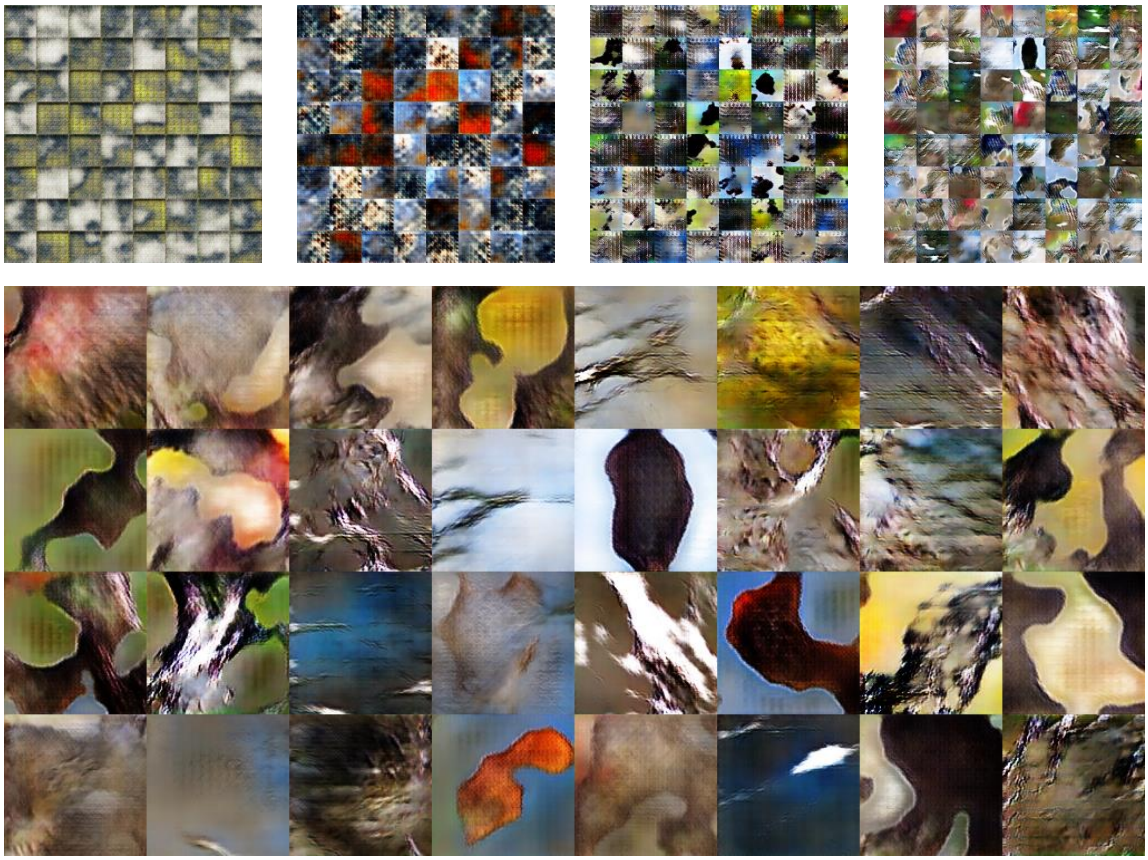
In order to make the game easy to access and share, it is built for the web. With some improvements to the code, it should be accessible for most devices and browsers. The current game is a first iteration, built quickly as a proof-of-concept to run on my personal computer.

In terms of improvements, I would want to make sure that the game is compatible with accessibility software such as screen readers. I would also like to make sure that the colour palette is suitable for people with different visual impairments. Through further playtesting, I would look for language that is hard to understand to make sure the game gets its point across clearly.

generating images

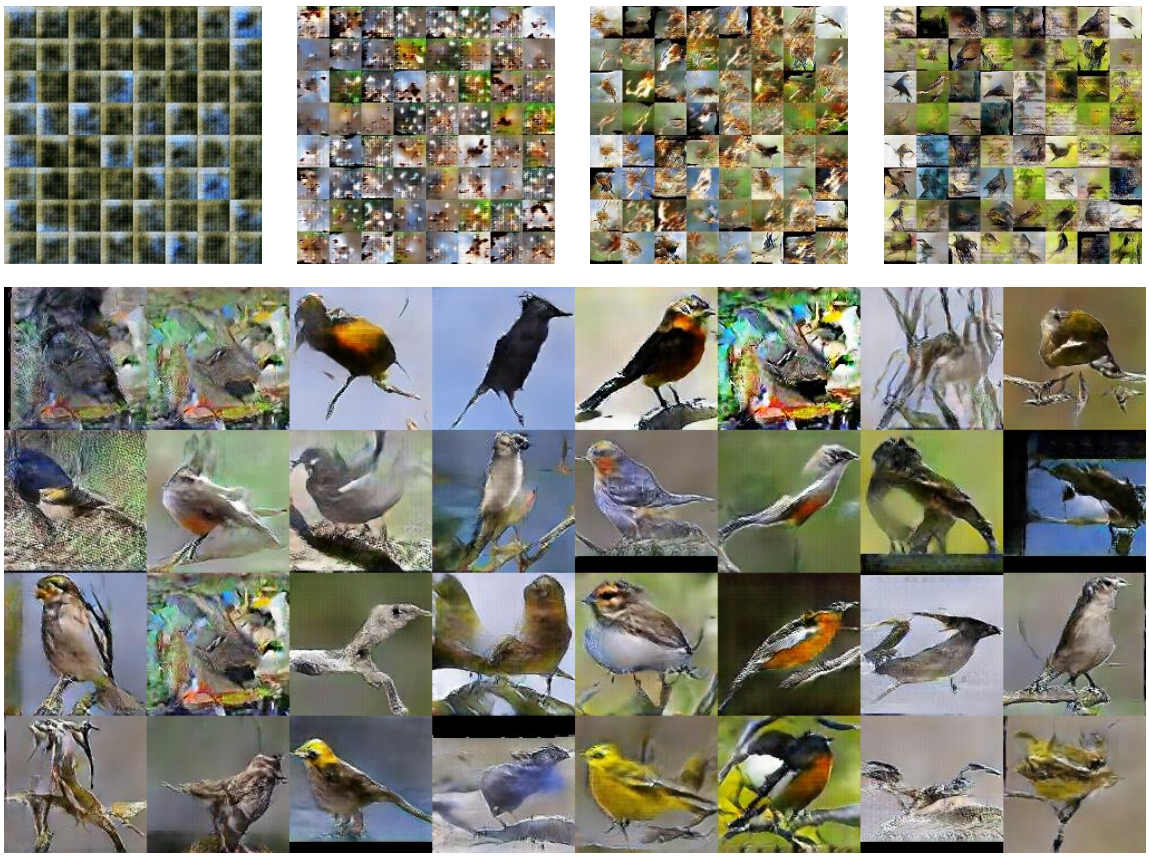
After using textgenrnn on Google Colaboratory, I decided to follow up on my bird name experiments with an attempt to generate images of birds. This turned out to be a lot more technically demanding, but I eventually got Taehoon Kim's DCGAN-tensorflow[iii] running on my computer. A lot of the technical difficulties revolved around tensorflow and DCGAN-tensorflow only working with certain versions of Python, as well as trying to make tensorflow use the GPU on my desktop computer.

After DCGAN-tensorflow was working with the sample dataset, I downloaded the Caltech-UCSD Birds-200-2011 dataset[iv] to train my model. I tried to run the dataset with some very interesting results.



1: Outputs from 200, 800, 2000, 3000 and lastly 4600 iterations. Last image is cropped.

The dataset contains 11.788 images of different birds, but the images are in different sizes. I came to realise that, as the DCGAN implementation requires images of the same size, it crops all images by default to a 128x128 square. The outputs of my first test were visually very interesting, but I assume that most of the cropped images that the model had trained on didn't even contain any birds. Luckily, the dataset also comes with text files with bounding boxes for where the birds are in the different images. I spent some time trying out Pillow[v], a Python library for image manipulation, trying to use the bounding boxes to crop the images myself. After a lot of trial and error, I managed to resize all the images to 128x128 pixels, with the birds in the centre and as much as the background as possible cropped out.



2: Outputs from 200, 1000, 2000, 2800 and finally 63.000 iterations. Last image is cropped.

I left this running for a long time, and this is probably the most interesting output I've had. However, two things were bothering me. Firstly, the 8x8 grid of 128x128 pixel images led to the individual images being very small. Making the images larger took a couple of tries, as simply using larger images in the same 8x8 grid caused my GPU to run out of memory. Luckily, changing to a smaller grid made it possible to use images of 256x256 pixels. Second were the black lines that appeared in the images due to an error in my image-cropping code which would sometimes try to include areas outside of the source image. So, I went back to my image cropping code and made some fixes. After running the model with the new images overnight, I realised that the black lines were still there and that I still had some errors to fix in my code. The final code I used is included in my submission (crop_imgs.py).

3: Outputs from 2000, 5000, 10.000, 50.000 and finally 95.000 iterations.

I use some of these images in my game. They are not quite as bird-like as the ones in the 128x128 pixel images, but I quite like how uncanny they look. I assume that when working with these larger images, the model would require more training to get the same result as with the smaller images. Because of all this trial and error, I haven't had the time yet to find out how well the model performs when using more time to train on these new images. If I had more time, it would be interesting to try to retrain the model and see how the output would differ, as well as letting it run for longer and see if it would improve or keep outputting images in the same style.

generating text

For generating bird names, I started out using Max Woolf's textgenrnn[vi] on Google Colaboratory. I spent some time figuring out how to use the software and debugging some errors related to formatting the input and certain characters causing the program to throw errors.

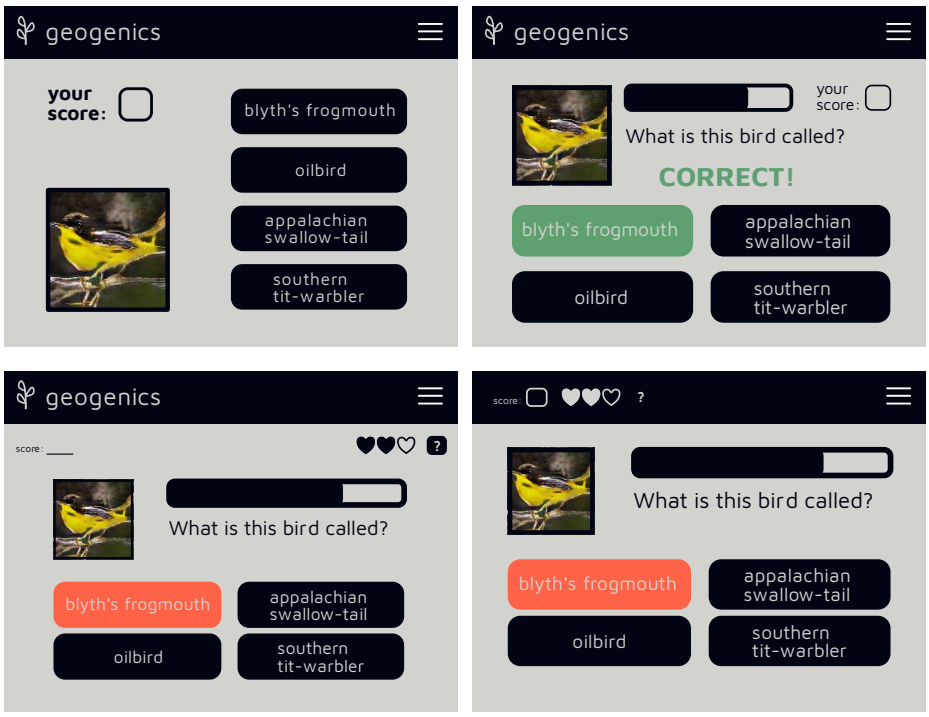
As the images were beginning to take form, I went back to text generation. Using Google Colaboratory was a good start, but the runtime environment would time out every now and then, making it hard to run in the background whilst working on other things. I downloaded textgenrnn to my computer and spent some time tweaking parameters so that the generated bird names would not just be copies of existing birds. Then I had the program save lists of generated names to file.

The next step was finding a way to compare these generated names to my list of real bird names to check if the generated names were truly unique. This really put my newly-learned Python skills to the test, but after hours of frustrated coding I finally managed to write a script which compared the list of generated names to the list of real names and returned whether the names were duplicates of real names, part real and part unique, or wholly unique. Armed with this knowledge, I was ready to start working on my game!

game design

Since I had decided to build the game for the web, I set up an express project using express-generator[vii]. In hindsight, it probably would have been simpler to use a framework like p5.js and p5.play, where you work with a set canvas size and a set of tools made for game development. Instead, I spent a lot of time working with css and figuring how to pass information between different parts of the web app.

I sketched out a rough outline of how I wanted the game to look and progress and designed a logo for the fictional company 'geogenics'. Then I made some mock-ups in a design software.



playtesting

I had two people play the game whilst giving me feedback as they played. I also asked them some questions afterwards. Sadly, I didn't have the time to make many changes after the playtest, but the feedback I got was valuable and would help me when developing the project further. Firstly, the game does not have a win-condition. The player plays until they make three mistakes and they are then taken to an 'epilogue' page talking about climate change. Based on the feedback, I would like to add another page between these which gives the player feedback based on their performance and ties the gameplay into the epilogue. I think this would tie the player experience together more nicely, instead of there being an abrupt ending.

There was also some confusion around the epilogue page having the exact same visual identity as earlier pages. The game is structured as an induction programme for new employees at geogenics, and the earlier pages are written as induction material produced by this fictional company. The epilogue, however, is written as me, the game designer. Based on this feedback, I have made some small, quick changes to the page, so that the two different voices are more visually distinct. Further playtesting would show how effective the change is!

A third piece of valuable feedback that ties into these two points was that the players did not necessarily see geogenics as 'evil' or bad. Thus, my idea of a morally dubious company is not communicated clearly enough in the current iteration of the game. The extra page I want to add before the epilogue could be used to communicate this as well, perhaps by insinuating that, based on your results from the game, they will exterminate certain species and replace them with these new artificial species.

There are also some technical aspects that I would want to improve on. In order for the timer to work in the game, the website refreshes every ten seconds. This also applies to pages that don't have actual gameplay, which causes an annoying refresh. It would probably be quite easy to fix, I simply ran out of time to fix it. Also, sometimes the web app skips the game and moves right to the epilogue. I would want to investigate why this happens, too.

I think that 'welcome to geogenics' is a game with an interesting core concept and fun gameplay. However, it is not a finished, polished project and it would benefit from more playtesting and iterative design.

inspiration

I have taken a lot of inspiration form the blog 'aiweirdness' by Janelle Shane[viii], who posts weird machine generated content which is often both hilarious and a really interesting look into how machines understand humans. I am also a big fan of Andy Wallace's 'Urza's Dream Engine'[ix], where he generated art from Magic: The Gathering playing cards using DCGAN-tensorflow.

The near-future setting of the game is influenced by sci-fi fiction such as the tv series Black Mirror[x] and the Pixar animated movie WALL-E[xi]. Geogenics was specifically inspired by Valve's portrayal of the 'over-the-top evil' high-tech company 'Aperture Science' in their Portal video games[xii]. 'Job Simulator' by Owlchemy Labs[xiii] also offers a funny, engaging and ridiculous parody of office jobs through a first-person simulator.

The game play takes cues from traditional arcade games and quiz games, with a visible score counter on the screen, hearts to show how many tries you have left until you lose the game, and the multiple-choice questions with four options, laid out like in game shows such as Who Wants to Be a Millionaire?[xiv].

resources

I have used the following resources in my project:

textgenrnn by Max Woolf
https://github.com/minimaxir/textgenrnn

DCGAN-tensorflow by Taehoon Kim
https://github.com/carpedm20/DCGAN-tensorflow

Norske navn på verdens fugler, Norsk Ornitologisk Forening
(Norwegian names for the birds of the world, The Norwegian Ornithology Association)
http://www.birdlife.no/fuglekunnskap/navn/

Caltech-UCSD Birds-200-2011
Wah C., Branson S., Welinder P., Perona P., Belongie S. "The Caltech-UCSD Birds-200-2011 Dataset." Computation & Neural Systems Technical Report, CNS-TR-2011-001.
http://www.vision.caltech.edu/visipedia/CUB-200-2011.html

express-generator by TJ Holywaychuk and Douglas Christopher Wilson
https://www.npmjs.com/package/express-generator

Pillow by Alex Clark and contributors
https://github.com/python-pillow/Pillow

Pure CSS Hamburger Fold-out Menu by Erik Terwan
https://codepen.io/anon/pen/pmdWLw?editors=1100

jQuery
https://jquery.com/

Fonts from Google Fonts
https://fonts.google.com/

+ all modules found in the 'node_modules' folder of my project.

All files in the public, routes and views folders in the turing-quiz web app are my own work, except where I have stated otherwise. The files crop_imgs.py and name_compare.py are also my own work. The birdnames.txt file is adapted by me from the list of birds from Norsk Ornitologisk Forening and the files of generated bird names were generated by me using textgenrnn.

how to run

The game should be available on http://www.doc.gold.ac.uk/www/178/. If that isn't working, you can run it locally if you have npm and node.js installed. In terminal, cd into turing-quiz and run 'npm start'. The site should then be available at localhost:3000.

[i] Janelle Shane 2017, *New paint colors invented by neural network*, last checked 24 May 2019, <https://aiweirdness.com/post/160776374467/new-paint-colors-invented-by-neural-network>

[ii] Dan Hon 2017, *I trained an A.I. to generate British placenames*, last checked 24 May 2019, <https://medium.com/@hondanhon/i-trained-a-neural-net-to-generate-british-placenames-9460e907e4e9>

[iii] See the list of resources

[iv] See the list of resources

[v] See the list of resources

[vi] See the list of resources

[vii] See the list of resources

[viii] https://aiweirdness.com/

[ix] Andy Wallace 2017, *Urza's Dream Engine & The RoboRosewater RoboDraft: Creating a machine learning algorithm to illustrate Magic cards*, last checked 24 May 2019, <https://andymakesgames.tumblr.com/post/167733819029/urzas-dream-engine-the-roborosewater-robodraft>

[x] Netflix 2011 – present, *Black Mirror*, tv series.

[xi] *WALL-E,* film produced by Pixar Animated Movies, California, (distributed by Walt Disney Studios Motion Pictures, 2008), DVD, 97mins.

[xii] Valve Corporation 2007, *Portal,* video game, PC, Valve Corporation, Washington.

[xiii] Owlchemy Labs 2016, *Job Simulator: The 2050 Archives,* video game, HTC Vive/PlayStation VR/Oculus Rift, Owlchemy Labs, Texas.

[xiv] Sony Pictures Television 1998 – present, *Who Wants to Be a Millionaire?,* tv series.