CARDIFF UNIVERSITY
School of Computer Science and Informatics

Using social media to observe wildlife distribution in the UK

CM3202 – One Semester Individual Project (40 Credits)
Final Report

William Dunn – c1317264
Supervised by Chris B Jones
Initial Plan Moderated by Ralph Martin
Final Report Moderated by Frank Langbein

Word Count: 24,966

Abstract

The purpose of this project is to identify if social media platforms, such as Flickr, can be employed to accurately represent existing wildlife behaviours in the UK, and recognise new, interesting, and undocumented species behaviours.

Species behaviour is currently observed via physically tagging species, or having dedicated enthusiasts and professionals recording sightings throughout the UK. The results of this project determine whether publicly available information can be utilised as a non-intrusive, and low cost method of wildlife observation and conservation.

Geotagged datasets are extracted from images available on social media to be utilised, in conjunction with Python scripts, to plot sightings between 2000 and 2017 for various classes of species. The datasets are plotted in full, on maps to determine each species' UK distribution, and are compared with a ground truth dataset extracted from the National Biodiversity Network Gateway. Similarity of social media and NBN datasets is determined by visually comparing map projections, and via standard similarity calculations such as F1 Score, R Squared and Hellinger Distance. Data sets are also in plotted in time slices to determine seasonal species behaviours such as migration, hibernation, and flowering.

The project concludes that wildlife behaviours can be observed for UK species via the use of social media. Certain species' distributions can be accurately identified using social media data when compared with the ground truth. While other species data are highly clustered in certain areas during certain months due to the behaviour of the general public and their social trends. Results suggest that the public tend to congregate at known breeding grounds during mating or birthing season for maximum chances of sightings and to view and photograph newly born species.

Acknowledgements

I would like to take this opportunity to thank all those who continuously shown support, dedication and interest in my education, project, and university career. Firstly, i'd like to thank my supervisor Professor Chris Jones for his keen interest, expert insight, and critical analysis of my report throughout the duration. Without his help, I would not have benefited from this project as fully as I have. I would also like to thank Padraig Corcoran for his interest and valuable insight which has helped to shape the report into its current format.

Next, I without a doubt need to thank my dedicated and loving parents and sister who have been there for me the whole way through university, and without who I would not be where I am today. Their care and critiques have been invaluable through academically the hardest years of my life.

Finally, I must thank all the friends whom I have made during my years at Cardiff University. My time would have been far less enjoyable without them, and university would not have been the life changing experience that it has been. Special thanks must be credited to the inhabitants of 36 Lisvane, and to the Cardiff University Lacrosse Team 2016/2017 who have made my final year the most memorable by far.

Contents

Abstract 1
Acknowledgements 1
Table of Figures 6
1 Introduction 8
1.1 Motivation 8
1.2 Aims and Requirements 8
1.3 Beneficiaries of Report 9
1.4 Assumptions 9
1.5 Section Summary 9
2 Background 11
2.1 Coordinate Systems 11
2.1.1 Longitude and Latitude 11
2.1.2 Easting and Northing 11
2.1.3 British National Grid and Irish Grid Reference 12
2.2 Citizen Science 12
2.2.1 National Biodiversity Network (NBN) Gateway: 13
2.2.2 Royal Society for the Protection of Birds (RSPB): 13
2.2.3 British Trust of Ornithology (BTO): 14
2.2.4 UK Butterflies: 14
2.3 Related Work 14
2.3.1 Can Geo-tags on Flickr Draw Coastlines? 14
2.3.2 Spatio-Temporal Sentiment Hotspot Detection Using Geotagged Photos 15
2.3.3 Using crowdsourced imagery to detect cultural ecosystem services: a case study in South Wales, UK. 15
2.3.4 Exploring place through user-generated content: Using Flickr tags to describe city cores. 16
2.3.5 Crowdsourcing indicators for cultural ecosystem services: A geographically weighted approach for mountain landscapes 16
2.3.6 Albatross numbers on remote islands are being counted via space 16
3.0 Methods 17
3.1 Wildlife Selection 17
3.2 Bird Species Selection 19
3.2.1 Resident Birds 19
3.2.2 Summer Birds 21
3.2.3 Winter Birds 22
3.2.4 Special Case Bird 24

3.3 Social Media Research 25
3.3.1 Twitter 25
3.3.2 Instagram 25
3.3.3 Flickr 26
3.4 Implementation Tools Research 27
3.4.1 Programming Language 27
3.4.2 Python Libraries 27
3.4.3 Pip 28
3.4.4 Conda 28
3.5 Analysis Techniques 28
3.5.1 Visual Analysis 28
3.5.2 Geo-Spatial Analysis 29
3.5.3 Quantitative Evaluation Methods 29
3.6 Method Selection 31
3.7 Dataset 31
4.0 Design 33
4.1 Software Development Cycle 33
4.3 Class Design 34
4.3.1 mapping 34
4.3.2 plotting 35
4.3.3 similarityCalculations 35
4.3.4 grid 36
4.3.5 geoConversions 38
4.4 Database Design 38
4.5 Key Scripts 39
4.5.1 FlickrDataCollection.py 39
4.5.2 createTimeline.py 40
4.5.3 processNBNData.py 41
5.0. Implementation 43
5.1 Flickr Data Collection 43
5.1.1 URL API Calls 43
5.1.2 FlickrAPI Python Module Implementation 43
5.2 Database 44
5.2.1 Creating Database Connection 44
5.2.2 Creating Tables 44
5.2.3 Inserting Data into Tables 44

5.2.4 Selecting Data 44
5.3 Plotting Maps 45
5.3.1 Creating A Map 45
5.3.2 Collecting Data to Plot 45
5.3.3 Plotting Data 46
5.4 Heatmaps 46
5.5 3D Mapping 46
5.6 Timeline 47
5.7 Grid 48
5.8 Evaluation Techniques 49
5.8.1 Generating Data Sets for Comparison 49
5.8.2 Calculations 49
5.9 Geo Conversion Methods: 51
5.9.1 Irish and British National Grid to Easting and Northing 51
5.9.2 Easting and Northing to Longitude and Latitude 51
6.0 Testing 52
6.1 Accuracy of Evaluation Implementations 52
6.1.1 R Squared Testing 52
6.1.2 Confusion Matrix 53
6.2 Accuracy of Coordinate System Conversions 54
6.3 Accuracy of Basemap Plotting 55
7.0 Results 57
7.1 Timelines 57
7.1.1 Flickr Geotagged Data 57
7.1.2 Seasonal Bar Charts 59
7.1.4 Timeline Evaluation 61
7.2 Grid Maps 62
7.2.1 Visual Comparison 62
7.2.2 Quantitative Evaluation of Results 66
7.2.3 3D Maps 69
7.2.4 Grid Map Evaluation 69
7.3 Map Projections 70
7.3.1 Time Slices 70
7.3.2 Dunlin Data 74
7.3.3 Map Projections Evaluation 74
8.0 Conclusion 75

9.0 Future Work 77
9.1 Computer Science Driven Future Work 77
9.2 Wildlife and Conservation Driven Future Work 77
10.0 Reflection on Learning 79
11.0 Appendix 80
11.1 - Figure 1: FlickrDataCollection.py algorithm 80
11.2 - Figure 2: Bar Chart Timeline Algorithm 80
11.3 - Figure 3: Line Graph Timeline Algorithm 81
11.4 - Figure 4: Data Conversion Algorithm 81
11.5 - Figure 5: Orange Tip Butterfly Timeline 81
11.6 - Figure 6: Grey Seal Map Projection 82
12.0 References 83

Table of Figures

Figure 01 Background: Longitude and Latitude Representation 11
Figure 02 Background: Universal Transverse Mercator Grid 11
Figure 03 Background: Irish Grid 12
Figure 04 Background: British National Grid 12
Figure 05 Background: NBN Graph Pheasant (right) NBN Example Output CSV (left) 13
Figure 06 Background: RSPB Map: Dunlin 13
Figure 07 Background: Create coastline based on highest adjacent cells (left) Large grid (middle) Small grid (right) 14
Figure 08 Background: San Francisco: Joy Plots 15
Figure 09 Background: South Wales Non-Urban Clusters 15
Figure 10 Background: NY: Downtown Tags 16
Figure 11 Background: Parameter Estimate Results 16
Figure 12 Research: KDE Example: Bandwidth Value-2 29
Figure 13 Research: Population Density Raster Grid 29
Figure 14 Methods: KL Divergence Discrete Probability Distributions Formula 29
Figure 15 Methods: Hellinger Distance Measure Theory Formula 29
Figure 16 Methods: Earth Mover's Distance Formula 30
Figure 17 Methods: Confusion Matrix Example 30
Figure 18 Methods: F1 Score Formula 30
Figure 19 Methods: Individual Species Dataset Sizes 32
Figure 20 Design: Agile Methodology Cycle 33
Figure 21 Design: Project Data Flow Diagram 33
Figure 22 Design: mapping Class Design 34
Figure 23 Design: Coordinate Variables Visualisation 34
Figure 24 Design: plotting Class Design 35
Figure 25 Design: similarityCalculations Class Design 35
Figure 26 Design: grid Class Design 36
Figure 27 Design: Grid Implementation Design 37
Figure 28 Design: Centre Coordinate Design 37
Figure 29 Design: geoConversions Class Design 38
Figure 30 Design: Database Design 38
Figure 31 Design: Activity Diagram - Flickr Data Collection 39
Figure 32 Design: Activity Diagram - Bar Timeline 40
Figure 33 Design: Activity Diagram - Line Timeline 40
Figure 34 Design: Activity Diagram - Convert British and Irish Grid Reference to Longitude and Latitude 41
Figure 35 Design: Data conversion data flow 41
Figure 36 Design: Activity Diagram - Plotting Data 42
Figure 37 Implementation: Flickr API URL Query Output 43
Figure 38 Implementation: 3D Mapping Example 47
Figure 39 Implementation: Grid Search Space 49
Figure 40 Results: 2000 - 2017 Total UK Geotags 57
Figure 41 Results: 2006 -2016 UK Geotags, Normalised 58
Figure 42 Results: 2006 -2016 UK Wildlife Geotags, Normalised, ... 58
Figure 43 Results: Atlantic Puffin Seasonal Bar Chart 2007 - 2017 59
Figure 44 Results: Common Bluebell Seasonal Bar Chart 2007 - 2017 59

Figure 45 Results: Canada Goose Normalised Data Flickr (right) NBN (left) 62
Figure 46 Results: Pheasant Normalised Data Flickr (right) NBN (left)..... 63
Figure 47 Results: Atlantic Puffin Normalised Data Flickr (right) NBN (left)..... 64
Figure 48 Results: Total Populated Cells per Species..... 64
Figure 49 Results: Wax Wing Normalised Data Flickr (right) NBN (left) 65
Figure 50 Results: Similarity Calculations 66
Figure 51 Results: Confusion Matrices..... 67
Figure 52 Results: Similarity Calculation Rankings..... 68
Figure 53 Results: Atlanitc Puffin 3D Map (left), Pheasant 3D Map (right) 69
Figure 54 Results: House Martin Time Slices 71
Figure 55 Results: Snow Bunting Time Slices..... 71
Figure 56 Results: Grass Snake Time Slices..... 72
Figure 57 Results: Common Frog Time Slices 73
Figure 58 Results: Common Bluebell Time Slices 73
Figure 59 Results: Dunlin All Data..... 74

1 Introduction

1.1 Motivation

Having always had a keen interest in wildlife and the outdoors I was very interested in pursuing a project that would allow me to combine my knowledge of computer science, acquired from my time at Cardiff University, with nature and conservation.

The continuous expansion of technology and the decline of wildlife can be regarded to have a direct relationship, in that, the destruction and pollution of wildlife habitats is consequential to powering the modern world. My motivation was to propose a project that would help to reduce some of the stigma of technology and demonstrate how wildlife can be appreciated and conserved with technology's aid.

While there are many examples of technology helping individual animals via the act of physical tagging, it is of particular interest to discover how technology can help to watch and monitor many animals in a less intrusive manner.

The use of social media is particularly interesting as the data, especially in terms of wildlife data, is typically supplied by enthusiastic members of the public who also have interests in wildlife. Using an image based social media platform is beneficial to conservation as it requires users to practice photography and to take a physical interest in the natural world. This in turn, via the sharing of photos, will hopefully inspire increasing amounts of the public using social media to capture their own photos and take steps to experience nature first-hand. Similar arguments are made to raise the profile of zoos, nothing can help encourage an interest in wildlife as effectively as physically experiencing it.

1.2 Aims and Requirements

The main aim of this project, as specified in the initial plan, is to extract data from geo-tagged images on social media platforms to be stored and queried in order to analyse the behaviours of UK wildlife. The intention is to find patterns that confirm existing knowledge of UK wildlife migration and hibernation, and also to discover some wildlife behaviours to help support the hypothesis that social media data is a viable data set for wildlife study.

As the project is research based, the original requirements were subject to be changed, due to its experimental nature, as and when new results were uncovered. The original aim was to analyse the data with the assumption that the extracted data accurately displays the wildlife distributions. However, as the project progressed it became less focused on what the data can uncover and increasingly directed towards determining if social media extracts can accurately display known behaviours in comparisons with ground truth, so that anything unknown will have greater value when noted.

Furthermore, as the project progressed interesting themes arose such as:

- How human population density can affect the results of the data?
- How the seasons can affect the amount of data being published to social media?
- How the public are attracted to specific known locations of certain species?
- Is social media a sustainable source for wildlife related geo data? Image based platforms are subject to huge dips in popularity when other platforms prosper.

All personal aims and objectives to improve my learning, and knowledge of computing techniques and wildlife have remained the same. My main topics of focus for personal growth with regards to computer science are, programming with Python and employment of geo-spatial computing.

1.3 Beneficiaries of Report

The audience for the project includes those who have an interest in either extracting and utilising social media data, geo-spatial computing, or wildlife conservation. While a front-end GUI tool won't be created and none of the scripts used in the project will be released to the public, the research results that can be used to support or contradict the use of social media to study wildlife within the UK can be found below.

Based on the result of the report, citizen science groups, such as NBN Gateway, could decide to utilise data extracted from social media in correlation with their own data sets, providing its users with higher quantities of data. In addition, if users of social media were aware that their photographs contribute to a charitable wildlife organisation, they may be more inclined to take more photos and include geo tags, despite privacy concerns, and hence expand the datasets.

The body of the project will contain information detailing how it is possible to extract and plot wildlife sightings. Conversely, this could have a negative impact if this knowledge were to be used maliciously to cause harm to wildlife.

1.4 Assumptions

- All photos taken from Flickr using full Common Name or the Latin Name are correct images of the species.
- There will be limits with regards to the amount of data that can be taken from Flickr.
- All data taken from NBN is accurate and can be used as a ground truth example.

1.5 Section Summary

Background: A description of the background of the project. The section covers the different coordinate systems that the project will use, several citizen science sites that will supply wildlife based knowledge to the project, and a description of projects that have completed tasks using social media geotags.

Methods: Extensive research to determine a diverse range of species with the aim of providing the best overall conclusion of the usefulness of social media data for studying UK species. Research of possible social media platforms to extract data from, the most appropriate programming language, the language libraries, the database type, and the mathematical and spatial analysis techniques that can be used to evaluate the successfulness of social media has been completed.

Design: A section detailing the design of the main components that are used to analyse the usefulness of Flickr data, and behaviour of the species. The data flow, classes, key script algorithms, and database are designed, depicted in diagrams, and explained.

Implementation: Implementation of each of the classes, the database, and the main components of the project such as data collection, plotting, and comparison techniques.

Testing: A section in which each of the results of the main components are tested to ensure they are correct. The results of the similarity calculations methods are tested against results of identical methods on different platforms. The accuracy of the data conversions from British and Irish grid to

longitude and latitude are tested to ensure they are correct, and the accuracy of the plotted data is tested.

Results: This section covers the most relevant results of the research. It covers the trends assessed from timeline analysis, the similarity calculations outcomes, and the results of visual comparisons of NBN and Flickr data. Additional interesting trends including the influence of species events (such as breeding) in engaging the publics' interest, measured in terms of photos taken. This section also evaluates all results.

Conclusion: A conclusion of the results of the project. It demonstrates how different times of the year and different events, such as breeding and birthing, affect the total number of photos. How some species are better represented by social media data than others, the decline of Flickr data as a research tool, and alternative uses of the results other than assessing the distribution of species in the UK.

Future Work: This section outlines a variety of possible expansions the project can undertake in the future, described in terms of either a computer science based approach or a nature and conservation based approach. Computer science approaches include, expanding the data set or the geographical scope of the project, updating the database to a geo spatial database, and performing further geo spatial analysis techniques. Nature and conservation based approaches include, public education using the datasets to best select locations and months to display educational information near local species, and further evaluating data that appears in Flickr but not NBN to determine if the uncharacteristic sightings are legitimate and if so, what event influenced the behaviour.

Reflection on Learning: This section encompasses personal reflection, highlighting achievements, areas for improvement, technical skills developed, and increased interest in wildlife and joy to have been able to complete a project that combined the two diverse fields of work.

2 Background

2.1 Coordinate Systems

A major requirement of this project is to plot and view wildlife distributions on a map projection. Therefore, it is vital to gain a thorough understanding of the coordinate systems that will be used to plot the data.

2.1.1 Longitude and Latitude

Longitude and Latitude are units used in the geographic coordinate system to represent its coordinates. Every point on earth has a specific latitude and longitude coordinate to help locate its position. Latitude at point P is the angle between the line perpendicular to the earth's surface at P and the plane of the equator. Longitude is the angle between the plane passing through P and the earth's axis, and the plane of the prime meridian. Both longitude and latitude are measured in degrees [1].

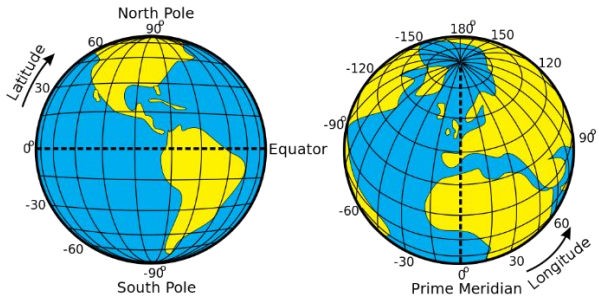


Figure 01 Background: Longitude and Latitude Representation

Longitude and Latitude are used for navigational purposes, locating your exact location, and even to evaluate the climate and local time of your location. In term of this project, longitude and latitude will be the primary coordinate system used to plot data from social media [2].

2.1.2 Easting and Northing

Easting and Northing are a geographic Cartesian coordinate system for any given point. Easting is the eastward-measured distance, or the x-coordinate. Northing is the northward-measured distance, or the y-coordinate. Easting and Northing are most commonly measured in meters from the axes of a datum, and are often associated with the Universal Transverse Mercator (UTM) coordinate system. Each grid in the UTM system will have identical easting and northing values, however their cell ID and hemisphere make the coordinate unique [3].

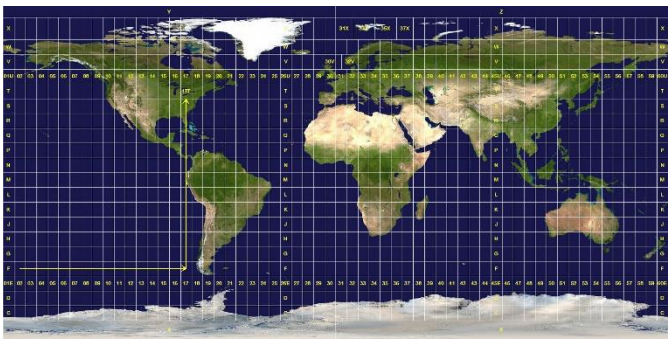


Figure 02 Background: Universal Transverse Mercator Grid

In this project, they are based on the Ordnance Survey National Grid reference system, also known as British National Grid (a system used by sites such as National Biodiversity Network Gateway).

2.1.3 British National Grid and Irish Grid Reference

The British National Grid and the Irish Grid are a system of geographic grid references used in Great Britain and Ireland respectively. The Irish Grid lies within the British grid however, it uses a different coordinate system better suited to its western position.

Both the British and Irish Grid systems use a 5x5 grid labelled A-Z (excluding I) from north west to south east. The British grid is identified as EPSG:27700 globally, and the Irish grid is identified as EPSG:29903 globally (these grid id's are used to convert any easting and northing values within the grid to the global longitude and latitude system.) [3]

The British grid is made up of 500km squares, divided into 5x5 grids of 100km squares each with two letters, furthermore the bottom left square of the grid (SV) is given a false easting and false northing value of 0,0 rather than 1000000, 500000. The Irish uses one grid of 5x5 100km squares each with a single letter.

Figure 03 Background: Irish Grid

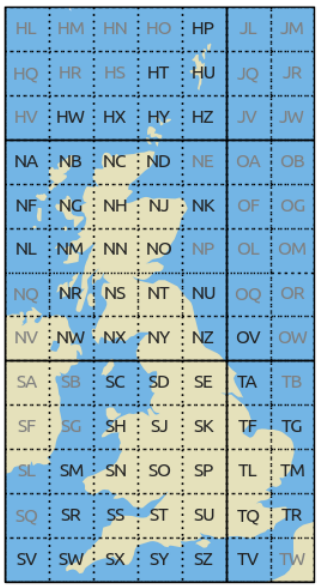


Figure 04 Background: British National Grid

The coordinates for this system can be displayed in two formats. The first is a grid reference that consists of the square, easting distance from square origin (bottom left corner) and, northing distance from the square origin. For example, G0305 means square G, 3km easting and 5km northing. This format commonly includes six digits to determine a 100m square location. The second method disregards the 5x5 squares and instead uses the origin of the grid (bottom left corner of V in Ireland, SV in Britain) and only uses the easting and northing coordinates. For example, 315904, 234671 is equivalent to O1590434671 within the grid. [4]

The easting and northing values can be converted to longitude and latitude be specifying the EPSG id of the grid, the gird id is relevant as easting and northing values will be repeated within each grid cell.

2.2 Citizen Science

Citizen science is the collection and analysis of data related to the natural world by amateurs in the public, usually as part of a collaborative project with professional scientists. [5]

Citizen science plays an important role within this project. It provides a useful reference knowledge source for all species researched, such as migration and hibernation times, distribution across the

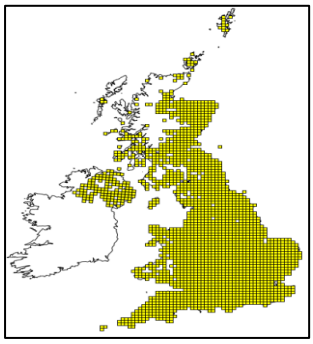
UK, and habitats, all of which can hopefully be reflected visually when plotted in time slices. More significantly location data taken from citizen science sites provides an excellent 'ground truth' data set which can be compared with extracted Flickr data to test its accuracy in displaying wildlife trends.

The examples below are not registered citizen science groups within the UK, however all are set up and contributed to by amateurs in the public and have relevance to this project.

2.2.1 National Biodiversity Network (NBN) Gateway:

The NBN Gateway is the web interface of the NBN used to search its extensive collection of data. The gateway has access to data from as far back as 1600, with over 1,300,000 million records from approximately 1000 different datasets, which is regularly expanded by up to 60,000 users. It is officially the largest collection of biodiversity information within the UK, and has revolutionised UK biodiversity data by allowing it to be shared, downloaded, analysed, and researched by the public. [6]

NBN has a feature that plots each of their specific species viewings on a map, in a similar format that the project uses to display the Flickr data. This is useful as it allows for comparison of the Flickr data set with the much larger and dominant NBN data set visually. The feature also allows specification of a time range for the data displayed on a UK map, and to download a CSV file describing each point and providing a specific grid reference. This presents opportunities for standard similarity calculations to be utilised. [7]



observati	recordKey	organisati	datasetKe	surveyKey	sampleKe	gridReference	precision	siteKey	siteName	featureKe	startDate	endDate
2866268	SEABIRD2	Joint Natu	GA000089	SEABIRD2	NBN-GA0	TA165750	100m	962075	Speeton	122067	16/06/2000	16/06/2000
2866259	SEABIRD2	Joint Natu	GA000089	SEABIRD2	NBN-GA0	TA175748	100m	962956	Trig Point	1651228	16/06/2000	16/06/2000
2866236	SEABIRD2	Joint Natu	GA000089	SEABIRD2	NBN-GA0	TA199742	100m	965125	Barlett Na	1289093	05/06/2000	05/06/2000
2866235	SEABIRD2	Joint Natu	GA000089	SEABIRD2	NBN-GA0	TA199742	100m	965125	Barlett Na	1289093	16/06/2000	16/06/2000
2866234	SEABIRD2	Joint Natu	GA000089	SEABIRD2	NBN-GA0	TA199742	100m	965125	Barlett Na	1289093	13/06/2000	13/06/2000
2866209	SEABIRD2	Joint Natu	GA000089	SEABIRD2	NBN-GA0	TA204737	100m	971528	Grandstan	1772919	26/05/2000	26/05/2000
2868002	SEABIRD2	Joint Natu	GA000089	SEABIRD2	NBN-GA0	SS129456	100m	968276	Lundy E	1284482	08/06/2000	08/06/2000
2868003	SEABIRD2	Joint Natu	GA000089	SEABIRD2	NBN-GA0	SS132460	100m	963538	Lundy F	1024880	08/06/2000	08/06/2000
2868022	SEABIRD2	Joint Natu	GA000089	SEABIRD2	NBN-GA0	SS129465	100m	971233	Lundy G2	1960662	05/06/2000	05/06/2000
2868033	SEABIRD2	Joint Natu	GA000089	SEABIRD2	NBN-GA0	SS130482	100m	963725	Lundy H	84687	05/06/2000	05/06/2000
2870144	SEABIRD2	Joint Natu	GA000089	SEABIRD2	NBN-GA0	HT962416	100m	971479	Foula 1	346977	05/06/2000	05/06/2000
2870146	SEABIRD2	Joint Natu	GA000089	SEABIRD2	NBN-GA0	HT972415	100m	970745	Foula 2	1542401	05/06/2000	05/06/2000
2870154	SEABIRD2	Joint Natu	GA000089	SEABIRD2	NBN-GA0	HT978391	100m	963686	Foula 4	1622874	05/06/2000	05/06/2000
2870189	SEABIRD2	Joint Natu	GA000089	SEABIRD2	NBN-GA0	HT951410	100m	962805	Foula 10	1484376	05/06/2000	05/06/2000
2870200	SEABIRD2	Joint Natu	GA000089	SEABIRD2	NBN-GA0	HT959392	100m	968021	Foula - inl	2245452	05/06/2000	05/06/2000
2873045	SEABIRD2	Joint Natu	GA000089	SEABIRD2	NBN-GA0	ND393855	100m	967877	Swona (Co	1208929	21/06/2002	21/06/2002
2873056	SEABIRD2	Joint Natu	GA000089	SEABIRD2	NBN-GA0	ND381838	100m	967308	Swona (Co	1847008	21/06/2002	21/06/2002

Figure 05 Background: NBN Graph Pheasant (right) NBN Example Output CSV (left)

2.2.2 Royal Society for the Protection of Birds (RSPB):

A charitable British organisation founded in 1889 to help promote the conservation and protection of birds. The website helps to identify birds and educate its users, providing useful information including migration patterns, location, descriptions, diet and more. [8] This information provides many opportunities for research, for example trends between birds and their food source, or time slice data confirming the migration patterns described. The RSPB website provides a UK map for each species of bird that is colour coded in accordance with the species residency within the UK, providing the opportunity to visually compare Flickr data plots to assess if the migration trends are similar.

Figure 06 Background: RSPB Map: Dunlin

- Green: Yearlong residents
- Orange: Summer residents
- Blue: Winter residents

2.2.3 British Trust of Ornithology (BTO):

A wildlife trust dedicated to providing knowledge on birds and their habitats. It functions via the help of enthusiastic volunteers who survey birds and provide unbiased information. [9] As well as providing information about birds they take an interest in all British wildlife and therefore provide a useful knowledge source for mammals, amphibians, and reptiles. The site provides useful information regarding the habitats and behaviours of species which will help to confirm that trends visible in Flickr data are legitimate.

2.2.4 UK Butterflies:

An awarded community for butterfly enthusiasts to share their knowledge and images of butterflies within the UK. The site consists of photo galleries, forums, reports, articles, and hundreds of pages worth of content describing every known British butterfly. [10] The information is useful for the project as it allows for the comparison of spatial and temporal data with the Flickr data set to further prove its worth.

2.3 Related Work

2.3.1 Can Geo-tags on Flickr Draw Coastlines?

(M Omari, M Hirota, H Ishikawa, S Yokoyama)

As the title suggest, the authors of this study created methods which allowed them to draw a coastline of the UK using just Flickr geotags. The coastline produced by their research was accurate within 500m of the actual coastline.

200 million geo-tagged photos were collected from Flickr using the tags ‘beach’, ‘sea’, ‘coastline’ and ‘shoreline’. It was noted that ‘beach’ retuned not only the most results, but the highest percentage of photos taken <500m from the coastline. They created the coastline by creating a grid and plotting the total count of photos within each cell, a line was then traced through all the highest populated adjacent grids. They applied the calculation of the coastline using two grids of varying size, the smaller the grid cells the more accurate the coastline was. See below for the visualisation of the grid drawing methods, the large cell grid results, and the small cell grid results:

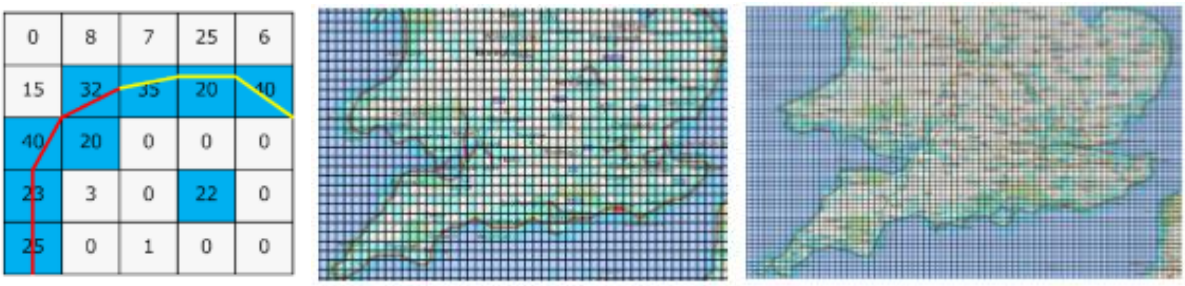


Figure 07 Background: Create coastline based on highest adjacent cells (left) Large grid (middle) Small grid (right)

Similar methods can be used within this project to calculate the accuracy of the Flickr data. If total counts are present in each cell of a grid covering the UK using both Flickr and NBN data, the grids can be compared to determine their similarity. However, one key difference is grid size. As it is unlikely that 200 million records will be collected and gird cells will be significantly larger to account for fewer geotags. [11]

2.3.2 Spatio-Temporal Sentiment Hotspot Detection Using Geotagged Photos

(*Y Zhu, S Newsam)*

An interesting report that has used geotagged photos to analyse public sentiment. The study looks for hotspots of emotion and shows how different emotions have different spatial distributions. It also utilised the capture time of photos to view year-by-year emotion hotspots, and compare this data with known events.

The authors decided to use Flickr as it provides a population's visual emotive reaction (via photos), which is arguably the most effective way to convey emotion. The study researches the occurrences of Ekman's six basic emotions, anger, disgust, fear, joy, sadness, and surprise in San Francisco, over a 10-year period. They study used a dataset of 1,753,903 images.

The study had no access to ground truth data, however the resulting hotspots makes logical sense. For example, locations, such as San Francisco Botanical Garden, and Alamo Square Park contained hotspots that expressed joy. Figure 08 displays all mentions of joy, it shows the two previously mentioned locations in red which signifies a hot location (blue equates to cold spots).

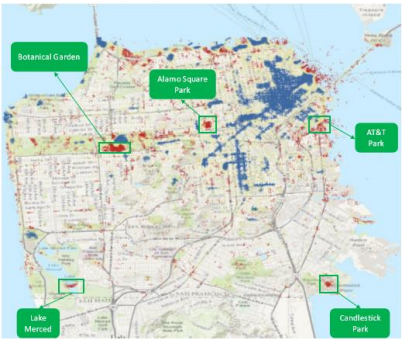


Figure 08 Background: San Francisco: Joy Plots

Within this project, similar colour schemes could be used to show areas of high and low concentration, for example an Atlantic Puffin breeding site would most likely be displayed as a hot location. It would be necessary to make changes to account for a significantly smaller data set. [12]

2.3.3 Using crowdsourced imagery to detect cultural ecosystem services: a case study in South Wales, UK.

(G Gilozzo, N Pettorelli, M Haklay)

A paper that aims to determine the reasons why and how people value certain places over others, and to measure cultural preferences associated with these areas. The authors assume that the appreciation of a place can be derived from the number of people taking and sharing photos of the area. South Wales has been used as a use case to identify geographic features of high cultural value.

A grid is used to display the total number of captured images in each cell, this is to visualise the areas of cultural significance. The study highlights non-urban interesting zones in red, and shows that area Rhossili Bay (the 2nd most popular beach in Wales) is most popular, and the 2nd most popular is Three Cliffs Bay (3rd most popular beach in Wales). [13]

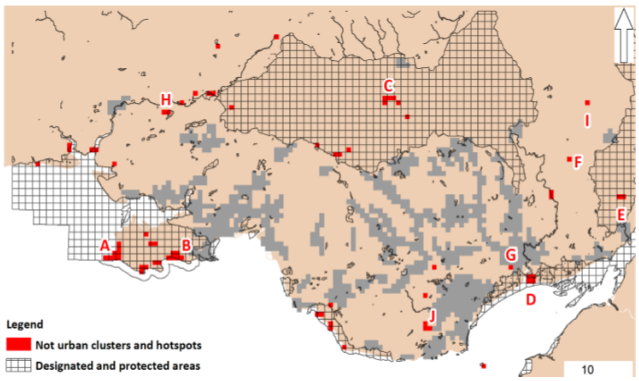


Figure 09 Background: South Wales Non-Urban Clusters

2.3.4 Exploring place through user-generated content: Using Flickr tags to describe city cores.

(L Hollenstein, R.S Purves)

Similarly to mapping the UK coastlines using tags such as beach, this study utilised metadata taken from 8 million Flickr photos to determine the borders of cities and neighbourhoods using keywords such as downtown. In addition, it seeks to uncover how people from cities across the USA geographically define terms such as downtown and other city core areas.

The authors used spatial techniques to plot data and calculate its accuracy, a documented test showing how many photos of 'hydepark' have geotags within Hyde Park. They also produced graphs on a city scale and USA scale that display the use of the word 'downtown', a prominent figure being one of Manhattan depicting the southern half as the city highly populated with the 'downtown' tag. Another interesting heat map figure displays areas in Chicago where the terms 'loop', 'downtown', and 'city' are used, the results are logical as the hot areas increase as the broader term is used. [14]

Figure 10 Background: NY: Downtown Tags

The results of this study are useful as they confirm that Flickr data can be used to successfully monitor human behaviours. It is likely that wildlife behaviours can also be monitored.

2.3.5 Crowdsourcing indicators for cultural ecosystem services: A geographically weighted approach for mountain landscapes

(P Tenerelli, U Demsar, S Luque)

A study that makes used of Flickr photos to determine how landscapes are associated with people's preference for cultural ecosystems in mountain landscapes. The authors of the study chose a small section of the Alpine Mountain Range and queried the Flickr API using a bounding box to return all geotagged images taken in the location between 2012 and 2014. They collected a total of 2174 photos, further reduced to 1326 after processing. The collected data was plotted within a grid and processed using geographically weighted regression and Moran's I measure of spatial autocorrelation. By plotting the data using a grid the authors could determine that crowdsourced data, such as Flickr data, can be utilised to identify spatial patterns of cultural ecosystem service preferences and their association with landscape setting. This is particularly useful to the project as a useful conclusion was made using a similarly sized dataset to this project's wildlife dataset. [15]

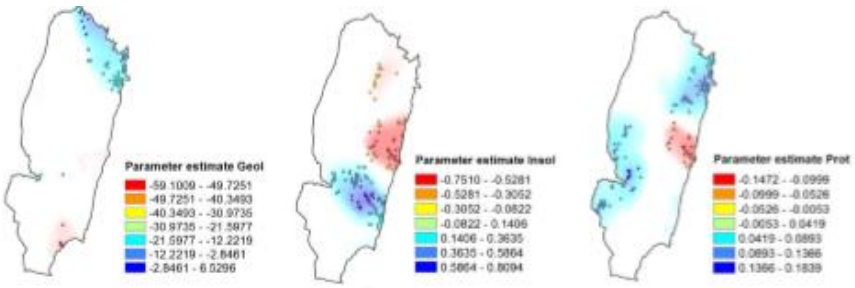


Figure 11 Background: Parameter Estimate Results

2.3.6 Albatross numbers on remote islands are being counted via space

An interesting development combining the fields of technology and wildlife conservation. The population of endangered Northern Royal Albatross are being counted and assessed from orbit using high resolution satellite cameras. This is a great example of how technology can help to monitor populations in a non-intrusive manner, ensuring that wild populations stay wild. [53] This study perfectly encapsulates the motivation of this project.

3.0 Methods

3.1 Wildlife Selection

The aim of this section is to help select a diverse variety of wildlife to study, that will return unique results and allow for a broader idea of Flickr's capability for studying wildlife. There is a significant variety of different vertebrates, invertebrates, and plant life, with different habitats and distributions in the UK. Therefore, it is important to ensure the majority of this diversity is researched.

Vertebrate Species:

Mammals: [9]

Species Name	Distribution	Habitat	Flickr Photos Available*
European Hedgehog (*Erinaceus europaeus*)	Widespread and abundant	Farmland and Urban Areas	Species Name: 6,604 *Latin Name*: 307
European Mole (*Talpa europaeus*)	Widespread and abundant	All areas, except high moors and mountains. Stay in tunnels most of the year.	Species Name: 7,641 *Latin Name*: 1 (many are not related to moles however)
Common Shrew (*Sorex araneus*)	Widespread and common	Farmland, woodland and hedgerows.	Species Name: 123 *Latin Name*: 29
Eurasian Otter (*Lutra lutra*)	West and South West England, and Scotland.	Rivers, estuaries, lakes, marshes, and coastal areas.	Species Name: 531 *Latin Name*: 1,711 (A significant number of photos originate from UK zoos)
European Pine Marten (*Martes martes*)	Common in parts of Scotland, very rare elsewhere.	Forested areas.	Species Name: 644 *Latin Name*: 1,359
Muntjac (*Muntiacini*)	Very common in parts of England. Originated in Asia and rapidly spreading.	Forests and wooden areas.	Species Name: 1,692 *Latin Name*: 2
Grey Seal (*Halichoerus grypus*)	Locally common on coasts of Scotland and South West England.	Temperate and Subarctic Waters.	Species Name: 15,077 *Latin Name*: 2,459
Red Fox (*Vulpes vulpes)*	Widespread and abundant.	Wooden area, Prairies, Farmland, and Urban Areas.	Species Name: 9,581 *Latin Name*: 4,826

Reptiles and Amphibians: [9]

Species Name	Distribution	Habitat	Flickr Photos Available*
Grass Snake (*Natrix natrix*)	Widespread and common throughout UK.	Tend to be most commonly found near water. Also near compost heaps for laying eggs.	Species Name: 3,279 *Latin Name*: 720
Slow-Worm (*Anguis fragilis*)	Located throughout UK.	Areas with access to sunlight, and thick vegetation.	Species Name: 1,330 *Latin Name*: 516
Adder (*Vipera berus)*	Commonly found in South England, Scotland, and Wales.	Heathland, Woodland, Moors. Requires sunny glades and slopes.	Species Name: 4,275 *Latin Name*: 1,277
Common Frog (*Rana temporaria)*	Common and abundant.	Semi-aquatic areas and water. Most commonly ponds.	Species Name: 3059 *Latin Name*: 1321
Great Crested Newt (*Triturus cristatus)*	Common and abundant	Semi-aquatic areas and water. Most commonly ponds.	Species Name: 370 *Latin Name*: 122

Birds: [8]

Species Name	Distribution	Habitat	Flickr Photos Available*
European Robin (*Erithacus rubecula*)	Resident throughout the UK.	Woodland, Hedgerows, Gardens.	Species Name: 2,135 *Latin Name*: 7,127
Eurasian Blue Tit (*Cyanistes caeruleus*)	Resident throughout the UK.	Woodland, Hedgerows, Gardens.	Species Name: 301 *Latin Name*: 2916
Atlantic Puffin (*Fratercula arctica*)	Coasts around the UK, Summer only.	Coasts and Islands.	Species Name: 2,196 *Latin Name*: 4,663
Redwing (*Turdus iliacus*)	Found throughout the UK in Winter. Parts of Scotland in Summer.	Hedges, Orchards, Fields.	Species Name: 4,304 *Latin Name*: 719
Common Kingfisher (*Alcedo atthis*)	Widespread in central and southern England.	Flowing water, Canals, Rivers, Coast.	Species Name: 2,029 *Latin Name*: 3,068

I**<u>nvertebrate Species:</u>**

Annelid, Arthropod, Mollusc: [9, 10]

Species Name	Distribution	Habitat	Flickr Photos Available*
Orange Tip Butterfly (*Anthocharis cardamines*)	Resident throughout the UK.	Hedgerows, Riverbanks, Woodland, Meadows.	Species Name: 2,915 *Latin Name*: 1,131
Honey Bee (*APIs mellifera)*	Resident throughout the UK.	Gardens, Woodlands, Orchards, Meadows (Trees surrounded by flowering plants)	Species Name: 3,882 *Latin Name*: 781
Garden Bumblebee (Bombus Hortorum)	Resident throughout the UK.	Gardens, Meadows (Areas with flowering plants)	Species Name: 3,561 *Latin Name*: 289

Plant Species: [16]

Species Name	Distribution	Habitat	Flickr Photos Available*
Common Bluebell (*Hyacinthoides non-scripta)*	Throughout Wales and England, less common in Scotland. Spring Flower.	Woodland	Species Name: 1,071 *Latin Name*: 2,491
Daffodils (*Narcissus pseudonarcissus*)	Throughout Wales and England, less common in Scotland. Spring Flower.	Woodland, Fields, Orchards.	Species Name: 38,928 *Latin Name*: 438

3.2 Bird Species Selection

As part of the research into the possibility of using the Flickr API data, ten bird species for the UK with different, yet clear, migration patterns were chosen evaluate using Flickr. Once plotted on a UK map, the output can be compared with existing knowledge from respected sites RSPB and NBN Gateway to assess the accuracy of using Flickr to evaluate known wildlife behaviour. [8]

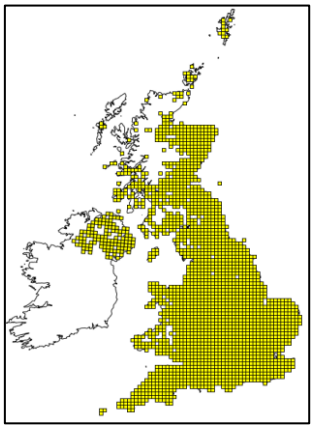
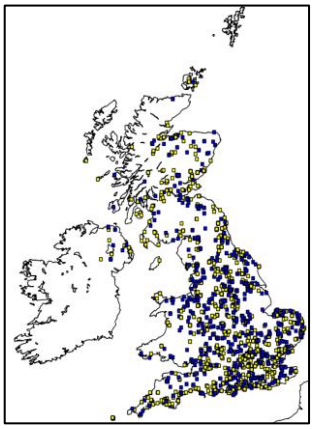
The graphs below for NBN and Flickr both display the full amount of data available from each source, notably, NBN provides a much larger data set in most cases. This data is not normalised in anyway at this point, therefore is not suitable for any kind of comparison.

3.2.1 Resident Birds

Residents describe all birds that stay within the UK throughout the year with no migration from the islands. Nearly every species of bird will migrate to some extent, be that from north to south of a country, however it is not so noticeable in the UK due to its small size. Birds in the USA that are considered residents will go as far north as Alaska on a seasonal migration.
For analysis of resident birds, the Pheasant, Canada Goose, and the Barn Owl have been chosen. These 3 birds are popular within the UK and therefore each returned a useful amount of data from API queries.

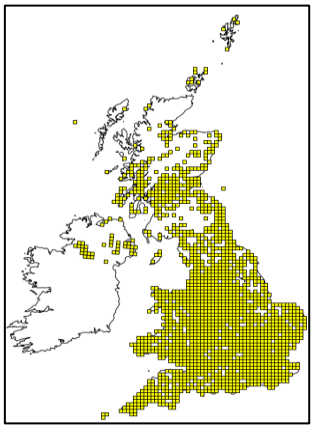
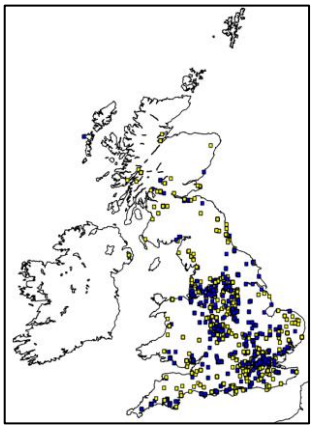
Pheasant (*Phasianus Colchicus*)

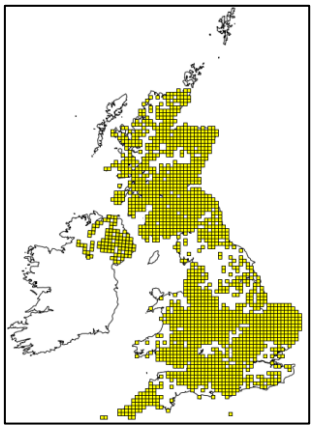
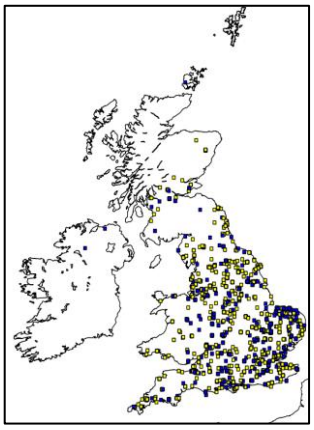
Brief Description: A large, long-tailed gamebird. Male are several shades of brown with black markings on the body and tail, with a recognisable green head and red face. The females are shades of brown all over, and lack the bright colours.
Location and Habitat: Common all year round across the whole of the UK, apart from the most northern areas of Scotland. Most commonly seen in open countryside nears woodlands. Uncommon in urban areas and upland.
Diet: Seeds, grain, shoots
Population: 2.3 million breeding females.

RSPB	NBN Gateway (110686)	Flickr API (3320)
		

Canada Goose (*Branta Canadensis*)

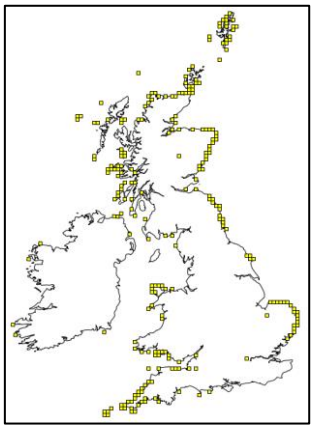
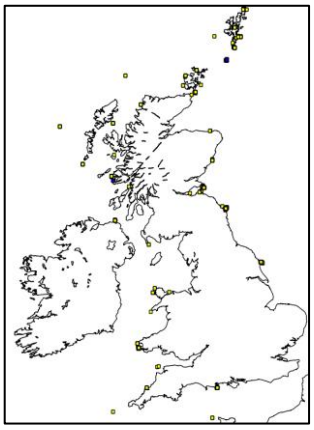
Brief Description: A large goose. Has a white and brown body, with a distinctive black neck and white throat.
Location and Habitat: Common year-round near lakes, gravel pits and town parks.
Diet: Roots, grass, leaves, seeds
Population: 124,000 breeding pairs. 190,000 birds during winter.

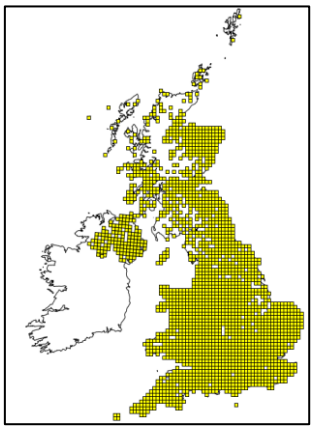
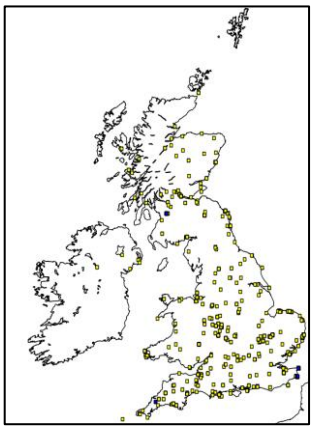
RSPB	NBN Gateway (93436)	Flickr API (2884)
		

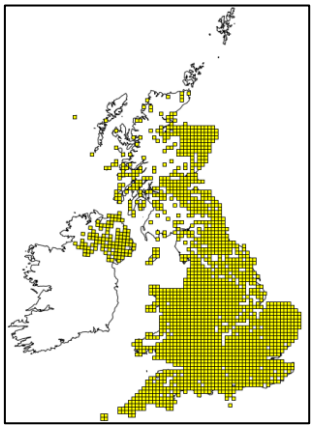
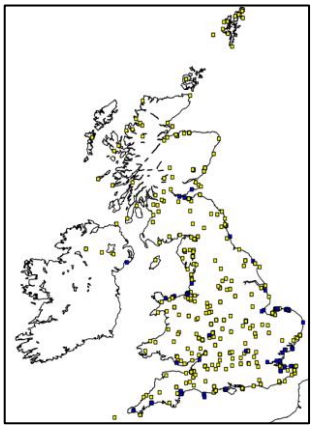
Barn Owl (*Tyto* Alba)		
Brief Description: White heart shaped face, brown back and wings, and white underparts. Location and Habitat: Viewable all year round during the day, most commonly seen at dusk. Located in open country, along field edges, riverbanks, and roadside verges. Diet: Mice, voles, shrews Population: 4,000 breeding pairs. 12,500 – 25,000 bird during winter.		
RSPB	**NBN Gateway (44348)**	**Flickr API (4385)**
		

3.2.2 Summer Birds

The term Summer species, describes all birds that stay within the UK in the summer months, usually to breed and hatch. For analysis of summer birds the Atlantic Puffin, Barn Swallow, and House Martin have been chosen. These 3 birds are popular within the UK and each returned a useful amount of data from Flickr API queries. They each have dissimilar habitats too, adding increasing variety to the project and something different to identify from the Flickr plots.

Atlantic Puffin (*Fratercula arctica*)		
Brief Description: A medium sized seabird with a black and white body, distinctive black head, white cheeks, and a colourful orange beak. Location and Habitat: Adults arrive in UK around March and April, and leave in mid-August. Can commonly be found in breeding colonies such as Skomer Island and Anglesey. Diet: Fish, particularly sandeels. Population: 580,800 breeding pairs.		
RSPB	**NBN Gateway (2463)**	**Flickr API (3231)**
		

Barn Swallow (*Hirundo rustica*)		
Brief Description: A small bird with blue backs, white underparts, and red throats. Location and Habitat: Swallows are seen in the UK March to October, in areas with an accessible supply of small insects. Diet: A range of small invertebrates. Population: 860,000 birds.		
RSPB	**NBN Gateway (99685)**	**Flickr API (807)**
		

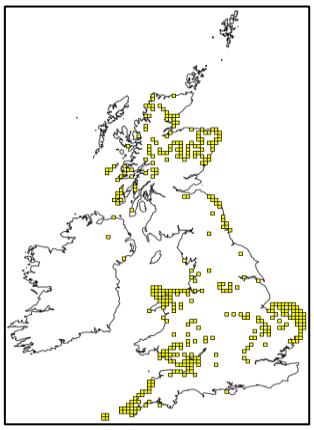
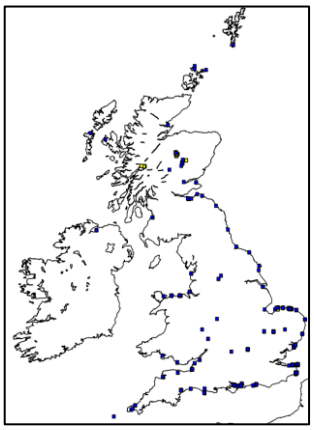
House Martin (*Delichon Urbicum*)		
Brief Description: A small bird with a blue-black back, and white underparts. Has a distinctive forked tail and white rump. Location and Habitat: Found across the UK from April to October, expect northern Scotland, and is found near man in towns and villages due to their diet of aerial insects found near farms. Diet: Insects Population: 510,000 breeding pairs.		
RSPB	**NBN Gateway (48748)**	**Flickr API (865)**
		

3.2.3 Winter Birds

The term Winter species, describes all birds that stay within the UK in the winter months, usually to feed and avoid colder weather in their breeding grounds. For analysis of winter birds the Snow Bunting, Brambling, and Wax Wing have been chosen.

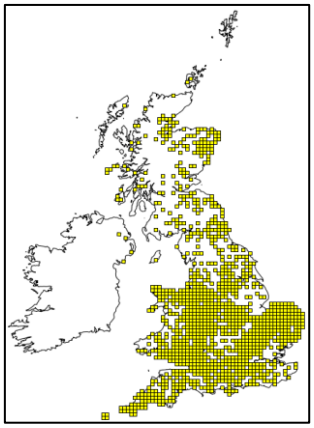
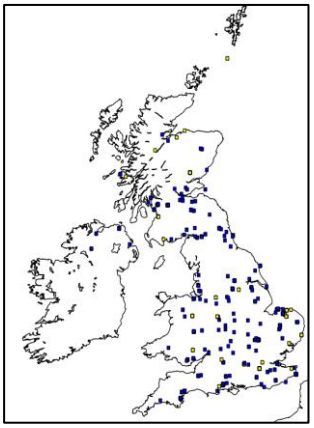
Snow Bunting (*Plectrophenax nivalis*)

Brief Description: A large bunting with a white head and underparts during the summer. They turn more brown during the winter.
Location and Habitat: Most commonly found in Scotland and coastal England as far south as Kent from late September to February/March. Spend summer in Scandinavia and North America.
Diet: Seeds and Insects.
Population: 60 breeding pairs. 10,000-15,000 birds during winter.

RSPB	NBN Gateway (5840)	Flickr API (599)
		

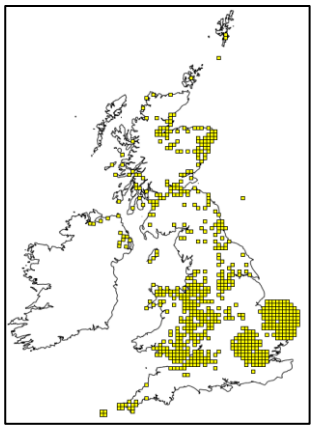
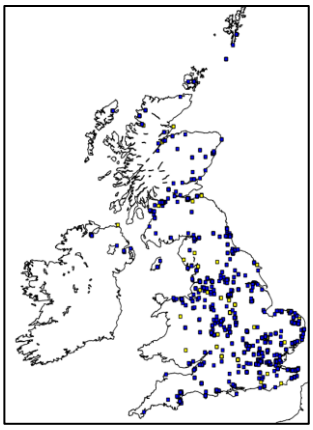
Brambling (*Fringilla montifringilla*)

Brief Description: A small bird with an orange breast, white belly, and brown-black back and wings.
Location and Habitat: Found in UK from September to March/April. Found in beech woodlands, farmland fields near woods, and gardens.
Diet: Seeds in Winter. Insects in Summer.
Population: 45,000 - 1,800,000 birds in Winter.

RSPB	NBN Gateway (17648)	Flickr API (650)
		

Wax Wing (*Bombycilla garrulous*)

Brief Description: A plump bird with a reddish-brown body, a black throat, and orange-yellow and black wings.
Location and Habitat: Found in the from October to March most commonly on the east coast of Scotland, but can be found inland due to the hunt for food.
Diet: Berries, especially rowan and hawthorn.
Population: 11,000 birds during winter.

RSPB	NBN Gateway (11012)	Flickr API (2590)
		

3.2.4 Special Case Bird

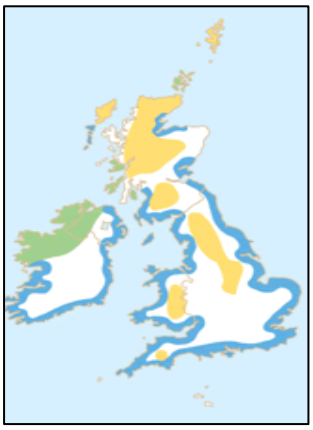
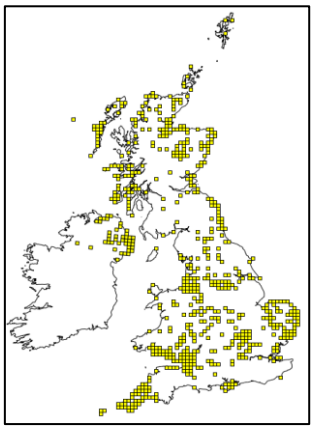
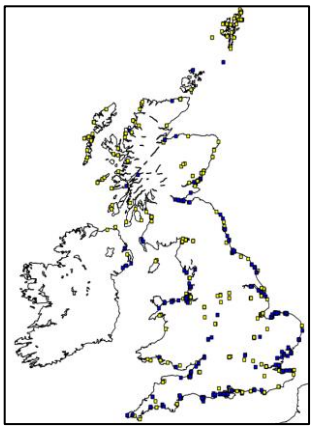
The Dunlin has been selected as a 'special case bird' as it migrates across the UK during the year. The migration of the species is very specific and therefore it would be interesting to see how well Flickr can represent the migration. See the RSPB graph below to see the seasonal locations.

Dunlin (*Calidris alpina*)

Brief Description: A small wading bird with brown body, white underparts and a distinctive black belly, and a curved black bill.
Location and Habitat:

- A yearlong resident in Northern Ireland.
- A winter resident around the coasts of the UK.
- A summer resident in central England, Scotland, and Wales for breeding purposes.

Diet: Insects, Snails, Worms.
Population: 8,600-10,600 breeding pairs. 360,000 birds during the winter.

RSPB	NBN Gateway (27939)	Flickr API (991)
		

3.3 Social Media Research

A description is provided for each of the three popular social media platforms that contain geotagged data. Based on the research below a suitable platform will be selected for this project.

3.3.1 Twitter

Background: Twitter is a social media platform used to broadcast short messages to the world as often as the user would like, it is about sharing opinions, images and hoping that others find your messages interesting or enjoyable. Alternatively, it can be used to follow and receive messages as users can create free accounts and have practically unlimited access to the opinions and messages of anyone they choose. [17]

As well as a social media site it is utilised as a platform for microblogging, a valuable platform for marketing, and one of the fastest means of sharing news globally. Information about an event can be tweeted almost instantly following its occurrence, skipping the process of writing and processing official news articles. Twitter has roughly 317 million active users each month. [18]

Available Geo-Tagged Information: Users of Twitter are given the option to enable location services on their accounts (which is switched off by default), this means they can add qualitative locations to their posts e.g. the name of a city or town. There is also an option to add a quantitative location utilising latitude and longitude to share an exact location. [19]

A study was performed using a weeks' worth of data, and the results revealed that 6% of users have enabled location services, and 20% of all tweets collected contained user locations specific to a street name or more precise location. Based on "internet live stat's" statistics of tweets 20% a week is equal to 700 million tweets. According to further studies 36% percent of tweets contain images. [20]

Therefore, there are roughly 252 million images tweeted with geo-tagged information every week, not including duplicates. This is clearly a very useful data source to consider for the analysis of wildlife. [21]

Available API:

Twitter has a very well documented API that allows you to query a portion off the data stream (<1% of total tweets) and receive data in JSON format. There are four main objects (Tweets, Users, Entities, Places). The Places object would be very useful for this project as it allows a user to collect geo data attached to tweets. The Places object is extensive and has a large range of attributes associated, which are also very well documented. [22]

Twitters API quota works in 15 minute windows, in which time a user can only extract a certain amount of data before having to wait for the next 15-minute window to extract more. The current restriction is 15 calls per application, however if multiple applications are used to extract data it can be removed faster (however additional OAuth codes will need to be requested). [23]

Several layers of authentication are needed depending on the use off the API, e.g. accessing locations requires greater levels of authentication than a typical Twitter developer.

3.3.2 Instagram

Background: Instagram is a social networking app that allows its users to shares photos and images from a smartphone. Each user who creates an account is given a profile from which they can upload and share photos, and a new feed from which they can view and receive posts from individuals that

the user chooses to follow. The app offers features such as in app image editing, sharing videos, following your favourite celebrities, and much more. [24]

Instagram is popular amongst both amateur and professional photographers to publicise their art, and to gain popularity via attracting followers. Businesses, big and small, also use Instagram to advertise products and increase their fan base. Instagram has a total of 600 million active users. [25]

Available Geo-Tagged Information: Users of Instagram can add a location to each of their posts as they choose. Once an image has been selected and edited a user can then choose an option to add a location to their image and once the image is posted they can edit or remove the location as they like. The locations are user defined and are usually qualitative names, such as a city or building name. [26]

On average 80 million images are uploaded to Instagram each day, that is 560 million a week. There is limited statistical information, with regards to geotagged image percentages available, however personal research from a frequent analyser of Instagram streams, estimates that roughly 31% of posts contain geotags. This is equal to 174 million images a week. [27]

Available API: Instagram has a well-documented API that allows a developer to request data. Data can be requested by using hashtags or by using a location id to return matching results. To get set up with the API a developer must register the application and describe the need for access to the API, next authentication is required, and finally the user can start to make requests to the API Endpoints. A user can access data using both a server-side or client-side application.

Like the Twitter API, Instagram has restricted the amount of calls an application can make within a 1 hour time window to 30 calls. [28]

3.3.3 Flickr

Background: Flickr is a site for photo sharing and hosting, it includes a range of advanced features to enable editing, tagging and subscribing to others (via RSS feeds if wanted). On Flickr, you can host hundreds of your own images for free, also there is a pro additional service that gives you unlimited storage and sharing for $2 a month.

Flickr is arguably more difficult to use than other image hosting sites, however its unique features and community of enthusiasts helps it to stand out. It is unique in that you do not organise your photos using titles or folders. Instead Flickr allows up to 75 tags to describe an image, you can also add notes and geotags to images. [29]

Available Geo-Tagged Information: Flickr offers functionality for any users to geotag their photos, be that manually or automatically from mobile phone / camera. A benefit of geotagging on Flickr is adding to the improvement of the search feature, as a user will be able to search for photos taken in a location rather than just a time or particular tag. Another benefit is improving likelihood of a user's image being located and liked online, hence adding to their popularity. There are numerous pages on Flickr actively pushing for more users to include geotags within their photos. [30]

In 2009 Flickr reached a milestone of 100 million photos containing geotags, out of a possible 3 billion photos that have been uploaded which equates to roughly 3.33% of all photos on Flickr. [31]

Available API: Flickr has an API available that is also very well documented, it also actively encourages developers to integrate their ideas with the API e.g. to help people make their photos available. To perform an action using the API a user needs to select a calling convention, send a

request to its endpoint specifying a method and arguments, and will then receive a formatted response.

An API key is needed as a parameter for a request. There are two types of API keys available, they are the Non-Commercial and Commercial Key. The Non-Commercial key is for apps that do not make money, or for people using the API to publish their own photos on their own websites. The Commercial Key is for users who want to make a profit. A Non-Commercial key is suitable for the needs of this project, and usefully does not require much validation to acquire. [31]

3.4 Implementation Tools Research

3.4.1 Programming Language

Based on web research, recommendations for supervisors, and personal preference it is decided **Python** will be used as the main programming language for this report. It is widely renowned for being a suitable choice when working with geographical information systems, it is simple to learn, requires less code to achieve similar tasks, you can wrap other programming languages such as C++, it is very well documented, and it includes a very large library of modules that can be imported.

3.4.2 Python Libraries

Basemap: Basemap is a toolkit that works in conjunction with Matplotlib. Basemap allows you to create a map projection in various formats, and convert longitude and latitude coordinates to a format Matplotlib can process. Matplotlib then plots its data on a figure, the map is then projected on the matplotlib figure. Basemap also includes several functions to facilitate the use of shapefiles, heatmaps, 3D mapping, and geographical projections of geographical features such as country borders and rivers. Basemap is very well documented online, with its own documentation and various forums and chatrooms providing support, which will be useful considering the time constraints of the project. [33]

CartopPY: A library with similar functionality to Basemap. It is designed to make drawing maps for data analysis and visualisation easier. Its key features are object orientated projection definitions, polygon and image transformations between projections, and powerful vector handling. Cartopy has out-dated Basemap in terms of functionality, however its documentations isn't nearly as comprehensible or well-presented. [34]

MatPlotLib: A library that produces publication quality figures. It can be used to create bar charts, error charts, timelines, scatter graph, and importantly geographic maps (via Basemap extension). Matplotlib is very well documented as it is regarded as one of the best Python libraries available.

FlickrAPI: A Python package used to access and extract data from Flickr. It requires the user to own an API key and a secret key to use, which can be requested from the Flickr website. Once the key is acquired you can easily run any of the Flickr API methods, such as Flickr.photos.search, and retrieve results in xml format.

PyMySQL: One of many packages SQL based packages in Python that allow you to query a SQL database. This is necessary as the results of FlickrAPI queries will be stored in a MySQL database.

CSV / XML: CSV and XML are both modules that allow you to parse through the filetypes by the same name. During this project, CSV is needed to parse through NBN extracts, and XML is needed to parse through Flickr extracts.

PyProj: A Python library for converting geographic coordinates (longitude, latitude) to map projection coordinates (x,y) and vice versa. This is will be used for converting NBN geotags to longitude and latitude, so that it can be plotted using Basemap. [35]

Mplot3D: A package used to create 3D objects. Specifically, an object called Axes3D which is used to create a 3D object on a 2D matplotlib figure. Using this object 3D bars can be plotted on a 2D map of the UK to better display volume of species is specific areas. [36]

3.4.3 Pip

Pip is a package management system that is used to install and manage many software packages written in Python. It is installed by default with installation of Python2.7.9 onwards. Pip can be used during this project to install many required packages. A huge advantage of Pip over the web based alternatives is that you can install almost any package with one line. [37]

3.4.4 Conda

Conda is also a package management system used to install and manage Python packages, however its main difference is that its open source. It has access to many useful user uploaded packages that Pip doesn't contain. Another huge advantage of Conda is it allows you to effortlessly create and switch between different Python environments. This is particularly useful when using packages like Basemap that don't work with newer versions of Python as you simply need to activate an environment that runs an older version of Python. [38]

3.5 Analysis Techniques

3.5.1 Visual Analysis

Map Projection Analysis: Visual analysis of a map projection of extracted Flickr data for specific species. This includes plotting all the data on a single map to view UK distribution, plotting time slices to view migrations and hibernation patterns, and plotting coordinates with seasonal colours to display seasonal variation. Map Projection Analysis also describes visually comparing the Flickr data projections with those created with ground truth NBN data to estimate its accuracy in displaying species distribution, and determining trends only viewable via social media data (such as viewing spots of certain species that are particularly popular with the public).

Timelines: Timelines are useful for visualising and analysing data from a large time period. They could be used within this project to visualise the total number of sightings of each species monthly for each year, to see the when the species is most commonly seen and if there are certain busy years. It can also be used to evaluate the total number of Flickr geotagged photos taken in general to see if the time of year affects total photos being taken, for reasons such as weather or holiday seasons (the results of this could justify normalising the Flickr data).

3D Map Projections: 3D maps can be created using Basemap and can help to provide a different view of spatial data. 3D maps can be used to display mountain ranges, a z axes for 3d columns, and once created in Python can be explored using the cursor. 3D maps are partially disputed due to their distortions often hiding key sections of data, e.g. a 3D map containing mountains ranges could hide data behind a peak.

3.5.2 Geo-Spatial Analysis

Kernel Density Estimation (KDE): A method used to estimate the underlying distribution of a random variable by calculating a weighted sum of a sample point near each location to be modelled. It can be utilised in geospatial computing to determine clusters of points, e.g. clusters of coordinates on a map. These clusters can then be created for monthly time slices to help determine a species migrating across the UK by visually analysing similarly positions clusters on sequential time slices.

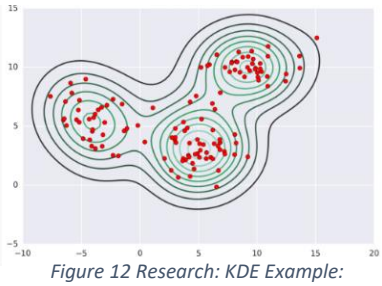


Figure 12 Research: KDE Example: Bandwidth Value-2

Automated Cluster Comparison: This follows on from utilising KDE alongside time slices to visually determine the movement of clusters. The key difference in this case is that geographically similar clusters from different time slices are automatically compared removing the need to human visual analysis. These methods are currently being researched by Dr Padraig Corcoran, Cardiff University.[50]

Raster Grid Modelling: Raster modelling involves dividing ground surfaces into regular grid cells, then a record is taken of the content of each cells. The content can be displayed in several ways, such as a colour theme or a count number. It is especially good for representing gradual change and imprecise phenomena, and poor for precisely surveyed objects due to its fixed resolution.

Figure 13 Research: Population Density Raster Grid

3.5.3 Quantitative Evaluation Methods

Kullback-Leibler (KL) Divergence: KL divergence is a measure of the non-symmetric difference between two probability distributions, part of the f-divergence class of statistical distance measurements. It can be used within the project to compare counts taken from Flickr and NBN data to achieve a measurement of the Flickr data's accuracy.[39] The KL Divergence formula is defined as follows, where P and Q are the two probability distributions:

$$D_{\mathrm{KL}}(P\|Q) = \sum P(i)\, \log \frac{P(i)}{Q(i)}.$$

Figure 14 Methods: KL Divergence Discrete Probability Distributions

Hellinger Distance: Hellinger distance is used to quantify the similarity between two probability distributions, it is also part of the f-divergence class of statistical distance measurements. It can be applied to the project in the same context as KL Divergence, to calculate the similarity of the NBN and Flickr datasets to find the accuracy of the Flickr dataset.[40] The Hellinger distance formula is defined as follows, where P and Q are the two probability distributions that are absolutely continuous with respect to third probability measure λ:

$$H^2(P,Q) = \frac{1}{2}\int\left(\sqrt{\frac{dP}{d\lambda}} - \sqrt{\frac{dQ}{d\lambda}}\right)^2 d\lambda.$$

Figure 15 Methods: Hellinger Distance Measure Theory Formula

Earth Mover's Distance (EMD): EMD is a measure of distance between two probability distributions over a specified region. It gets its name by the interpretation of calculating two different ways of piling dirt over a region, however it is also known as the Wasserstein metric. EMD is the minimum cost of transforming on probability distribution into the other. [41] The EMD formula is as follows, where P is a probability distribution (Flickr) that will be transformed into probability distribution Q (NBN), m and n are the number of clusters in P and Q respectively, $d_{i,j}$ is the ground distance between clusters p_i and q_i, and $f_{i,j}$ is the flow between p_i and q_i:

$$EMD(P,Q) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{i,j}d_{i,j}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{i,j}}$$

Figure 16 Methods: Earth Mover's Distance Formula

Coefficient of Determination (R^2): A number which when calculated can be used to calculate future results or test a hypothesis. The R^2 is simply the square of the correlation coefficient (r) between true outcomes and predicted outcomes, in the case of this project it would be between NBN and Flickr. The R^2 value ranges from 0 to 1.

Confusion Matrix: A confusion matrix is used to describe the performance of a classifier on a test data set for which the true values are known, Flickr being the test data set and NBN the true values in terms of the project. A confusion matrix at its most basic form is a 2x2 table where each square represents a different relation between the test and true data. [42] The relations are:

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Figure 17 Methods: Confusion Matrix Example

- True Positive: True data and test data agree there is a species in a location.
- True Negative: True data and test data agree there isn't a species in a location.
- False Positive: Test data states there is a species in a location, true data does not.
- False Negative: True data states there is a species in a location, test data does not.

Recall: Recall is calculated by dividing the number of True Positives by the True Positives plus the False Negatives. It is a standard measure of similarity used to state how many of the true data points (NBN) are also displayed by the test data points (Social Media). If there are 10 points in the true data, and the test data displays 10 of the 10 points then the recall would be 100% or 1.0.

Precision: Precision is calculated by dividing the number of True Positives by the False Positives plus the True Positives. In the previous example 10 points were correctly displayed by test data, however there may be an additional 5 incorrect points that are not represented by the true data. This means that although 100% of events are recalled, the precision is only 66% or 0.66.

F1 Score: F1 Score is calculated using recall and precision. It is used because precision and recall alone are not an accurate representation of one data set's superiority over another, as one could have better precision and the other a better recall. The F1 Score provides an harmonic mean that gives a clearer view of data sets accuracy when compared with ground truth. [51]

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

Figure 18 Methods: F1 Score Formula

Accuracy: Accuracy is calculated by dividing the number of True Positives plus True Negatives by the total number of data points in the confusion matrix. It is used to give an overall indication of how often the test data set values are correct.

3.6 Method Selection

In this section, you can see the selected tools for the project based of the results of the research displayed above.

Topic	Selection
Species:	Atlantic Puffin, Common Bluebell, Common Frog, Eurasian Otter, Fallow Deer, Grass Snake, Grey Seal, Honey Bee, Orange Tip Butterfly, Red Fox.
Bird Migration:	Atlantic Puffin, Barn Owl, Barn Swallow, Brambling, Canada Goose, Dunlin, House Martin, Pheasant, Snow Bunting, Wax Wing.
Social Media:	Flickr
Python Mapping Library:	Basemap
Database:	phpMyAdmin MySQL
Analysis Methods:	Timelines, Map projections, 3D mapping, raster grid, Hellinger, coefficient of determination, confusion matrix.

3.7 Dataset

The dataset will be established using data extracted from Flickr, a social media platform. Flickr was selected as the geotags should provide a coordinate accurate to where the species sighting took place. This is due to Flickr requiring you to upload an original photo therefore the user must have identified the species themselves. Platforms like Twitter are heavily opinion based and geotags could have no relation to the topic of the tweet and its attached tags.

Based on Flickr data being generated via the public it is assumed that certain facts may affect the data. Such as the higher populated areas in the UK uploading higher number of photos, the weather affecting the amount of data available for certain months of the year, and known wildlife havens within the UK generating higher numbers of wildlife species.

There is also the possibility of a species being incorrectly tagged by an amateur photographer causing certain species datasets to have outliers, and smaller datasets having distortions caused by an individual photographer uploading multiple photos of the same species sighting causing graphs and map projections to appear as though many sightings have occurred in an area with relatively low wildlife. Finally, there is also the issue of less popular or unknown species having Flickr datasets too small for effective research. Thus, people are much more likely to correctly identify and upload a photo of a large mammal (e.g. Red Fox) than a small invertebrate (e.g Megopthalmus scabripennis).

The data will be extracted from Flickr using the Common Name and Latin Name of the species to help provide a larger, more accurate data set. As the title of the project suggests, data will only be extracted for the UK. To extract data there are two options. The first is to use a bounding box covering the British Isles, this causes issues as it will include part of Ireland and some of France (which isn't in the scope of the project), or will require multiple bounding boxes to be used (which again won't ensure only the UK will be in scope). The second option is to use the Flickr API place_id parameter that allows you to select geotags from a specific location, this is ideal as using the place_id 'cnffEpdTUb5v258BBA' only geotags from the UK are searched. Using place_id has greater accuracy and simplicity.

See below for the summary of the project dataset.

Species	Flickr Count	NBN Count
Atlantic Puffin	3,231	2,486
Common Bluebell	2,413	N/A
Common Frog	752	N/A
Eurasian Otter	515	N/A
Fallow Deer	3,601	N/A
Grass Snake	592	N/A
Grey Seal	3,119	N/A
Honey Bee	2,956	N/A
Orange Tip Butterfly	536	N/A
Red Fox	2,038	N/A
Barn Owl	4,385	44,787
Barn Swallow	807	101,658
Brambling	650	17,652
Canada Goose	2,884	93,687
Dunlin	991	28,741
House Martin	865	49,511
Pheasant	3,320	112,126
Snow Bunting	599	5,847
Wax Wing	2,590	11,031
Total	36,844	467,526

Figure 19 Methods: Individual Species Dataset Sizes

4.0 Design

4.1 Software Development Cycle

The software development cycle that the project will follow, is the agile development model. Agile is an iterative development cycle in which requirements and solutions will change with each iteration, over the duration of the project. Due to this project being experimental with a research based approach, agile is very useful due to its flexible nature. If new knowledge is uncovered because of research, the new knowledge can be investigated in the next and future iteration, if interesting results are uncovered. The project will work in weekly iterations. At the end of each week, meetings will take place to evaluate the week's results, and to define new developments for the next week.

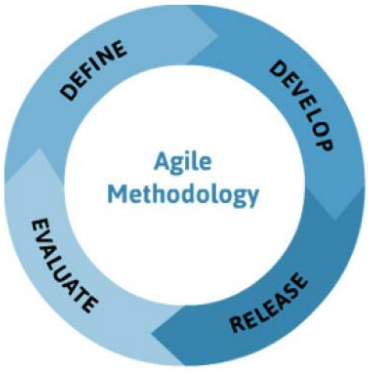


Figure 20 Design: Agile Methodology

4.2 Data Flow Diagram (DFD)

A DFD demonstrates how an information system processes data. It illustrates the input and outputs of the data, including where the data originates, how it is used, where it goes, and how it gets stored. It is useful to help describe the boundaries of the system, communicating the existing system to others (both technical and non-technical), and it can help support its designer's logic behind the systems dataflow.

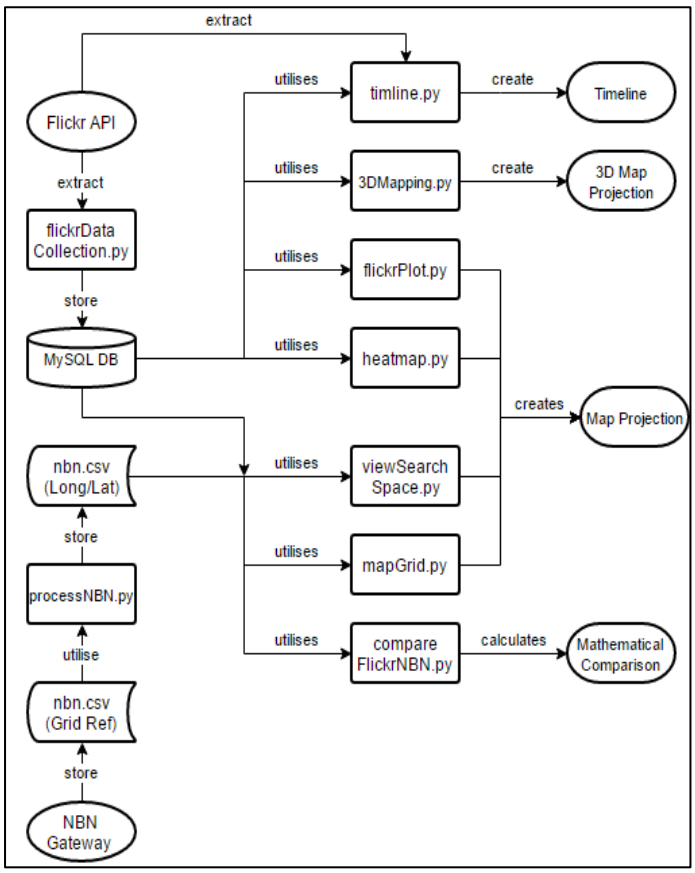


Figure 21 Design: Project Data Flow Diagram

Figure 20 displays the designed data flow for the project. There are two data sources, the Flickr API and the NBN Gateway. The Flickr data will be extracted using a Python script and then stored in a MySQL Database. The NBN data will be extracted manually from the Gateway as a CSV file, then the grid reference coordinate system used by NBN is converted into longitude and latitude.

The Flickr API will also be directly used by timeline.py to create a timeline that displays the total number of Flickr tags in the UK. The purpose of this is to evaluate the future worth of the Flickr dataset.

The MySQL and NBN data are then utilised by a variety of Python scripts to produce a variety of outputs, such as a Timeline, Map Projection, and similarity calculations such as R-Squared, and F1 Score.

4.3 Class Design

This section will detail the classes designed to provide functionality that would frequently be needed during the research. The classes contain methods to create maps, plot and compare data, to create a flexible and manoeuvrable grid, and for any data conversions needed for geo data. Each of these classes will be stored in a python file to be accessed by each implemented python script.

4.3.1 mapping

mapping
+lowLat: int +highLat: int +leftLon: int +rightLon: int
+ Basemap createMap(int lowLat, int highLat, int leftLon, int rightLon, String shapefile = 'no', String river ='no'):

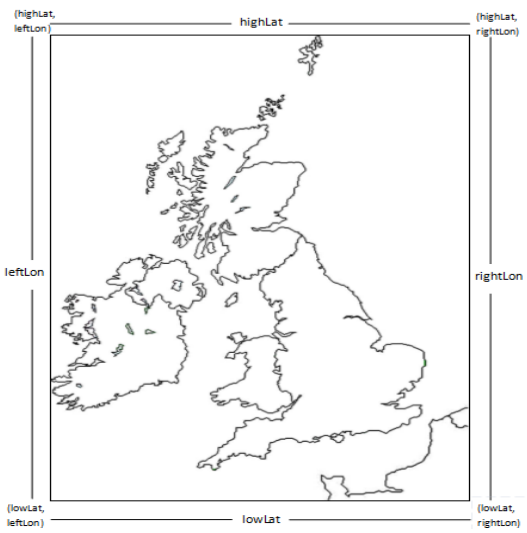
Figure 22 Design: mapping Class Design

int lowLat: Defines the value of the lowest latitude value for the desired map projection.
int highLat: Defines the value of the highest latitude value for the desired map projection.
int leftLon: Defines the value of the left longitude value for the desired map projection.
int rightLon: Defines the value of the right longitude value for the desired map projection.
Basemap createMap(): Takes longitude and latitude coordinates as parameters and returns a Basemap map projection. There are additional optional parameters to display shapesfiles, or to display rivers on the map projection.

The purpose of this class is to create a map projection of the UK which can then be used to plot data using the plotting class. This class is used as the basis for almost all scripts within the project, such as providing a background map for the raster grid, time slice plotting, visual comparison of NBN and Flickr data.

The optional parameters are by default set to not be used, however under certain circumstances they can be useful. For example, the shapefile displays each of the counties within the UK and can be used to more clearly to see if certain species are attracted to human populated areas. An example of how the river parameter can be used is to determine if water based species, such as Grass Snake or Eurasian Otter, are more commonly found near their river habitats as initial habitat research would suggest.

The coordinate parameters are used as depicted in figure 21. You can see that the four parameters are each used in two corners to help define the complete bounding box of the map projection. The project will only cover wildlife distributions within the UK, therefore the values for highLat, leftLon and rightLon will be fixed. The value for lowLat is subject to change depending on the size of the grid. Further detail on this design can be found in the implementation section.



4.3.2 plotting

plotting
+getSQLData(String columnName, String tableName, int minYear, int maxYear, int minMonth, int maxMonth) +getCSVData(String filepath) +processSQLOutput(sqlOutput)

Figure 24 Design: plotting Class Design

getSQLData(): Takes column name and table name as parameters in order to locate the correct data, e.g. longitude, Atlantic_Puffin. The method also requires input of a min and max year, and a min and max month, allows the user to specify a time range for the data being searched. This method will return an array that contains the SQL output, and the amount of files returned.
getCSVData(): Takes a file path as a parameter. The parameter must be a file path to the CSV that the user wishes to retrieve data from. The method will return an array full of latitude/longitude values.
processSQLOutput(): Takes the output of getSQLData() as an input. It uses regular expressions to remove the excess data from the results, such as brackets and colons. It will then return an array of values, e.g. longitude values.

The purpose of this class is to provide methods which assist plotting of extracted data. The class is used in many scripts throughout the research, for example to plot time slice data, display full datasets on single projections, and to compare NBN and Flickr data visually.

The year and month parameters within getSQLData() were included to add flexibility when searching for data. By including those parameters, searches for decades of data or for months can be done depending on the requirements of the agile iteration.

The processSQLOutput() was included to convert the following SQL input:

((Decimal('-0.614722'),), (Decimal('-1.950073'),), (Decimal('-3.001327'),), (Decimal('-0.497817'),), (Decimal('-2.554836'),), (Decimal('-1.055459'),), (Decimal('0.091838'),))

Into the following useful values.

[-0.614722, -1.950073, -3.001327, -0.497817, -2.554836, -1.055459, 0.091838]

4.3.3 similarityCalculations

similarityCalculations
+hellingerCalculation(array sqlPD, array csvPD) +KLDCalculation(array sqlPD, array csvPD) +rSquaredCalculation(array sqlCountArray, array csvCountArray) +confusionMatrix(array sqlCountArray, array csvCountArray) +precisionCalculation(int truePositive, int falsePositive) +recallCalculation(int truePositive, int falseNegative) +accuracyCalculation(int truePositive, int trueNegative, int total) +f1Calculation(int precision, int recall)

Figure 25 Design: similarityCalculations Class Design

hellingerCalculation(): Used to calculate the Hellinger distance of two probability distributions.
KLDCalculation(): Used to calculate the Kullback-Leibler divergence of two probability distributions.
rSquaredCalculation(): Used to calculate the coefficient of determinations between two data sets.
confusionMatrix(): Used to create a confusion matrix for 2 data sets.
precisionCalculation(): Used to calculate the precision of a confusion matrix using the results of confusionMatrix()
recallCalculation(): Used to calculate the recall of a confusion matrix using the results of confusionMatrix()
accuracyCalculation(): Used to calculate the accuracy of a confusion matrix using the results of confusionMatrix()
f1Calculation(): Used to calculate the f1 score using results of precision() and recall().

The purpose of these methods is to compare Flickr data with NBN (Ground Truth) data. Multiple calculations have been selected to determine that each calculation returns similarly ranked results, and to ensure provide further clarity of the accuracy of the Flickr data.

The parameters SQLPD, csvPD, SQLCountArray, and csvCountArray are arrays where each value corresponds to the total count of a certain species found within a specific region/cell. The cells are defined using the grid class. The data in these arrays has been normalised due to the significantly different size of the Flickr and NBN dataset, the normalisation process converts each count into a percentage between 0-100, and then rounds each percentage to the closest 0.5% (hence removing very small NBN values).

4.3.4 grid

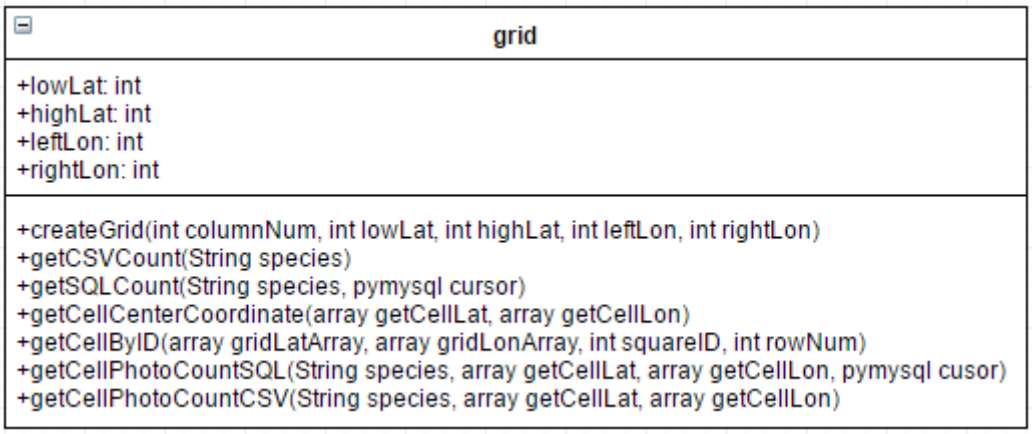


Figure 26 Design: grid Class Design

int lowLat: Defines the value of the lowest latitude value for the desired map projection.
int highLat: Defines the value of the highest latitude value for the desired map projection.
int leftLon: Defines the value of the left longitude value for the desired map projection.
int rightLon: Defines the value of the right longitude value for the desired map projection.
createGrid(): Creates an array of longitude and an array of latitude values which when plotted will display a grid. The size of the cells are updated by increasing/decreasing the columnNum value.
getCellByID(): Input the ID of a cell, the method will then search the grid and return the coordinates of the cell in an latitude array and a longitude array.
getCellCenterCoordinate(): Takes a latitude and longitude array which contains the coordinates of a cell. It will then calculate the exact centre of the cell and return the longitude/latitude coordinates.
getCellPhotoCountSQL(): Takes a latitude and longitude array which contain the coordinates of a cell. It will then determine how many SQL records for a specific species are found within the cells

bounding box.
getCellPhotoCountCSV(): Takes a latitude and longitude array which contain the coordinates of a cell. It will then determine how many CSV records for a specific species are found within the cells bounding box.
getCSVCount(): Get the total number of longitude/latitude values within a CSV file.
getSQLCount(): Get the total number of values from a database table.

This class was implemented to easily create and search a grid for purposes of visual comparison, improved visualisation of density in a specific area, and to aid with similarity calculations. It is useful for analysing both NBN and Flickr data.

The grid is created following a designed algorithm. It works by first collecting the coordinates of what will be the top right and top left corners of the map projections, next you calculate the width of the cells by dividing the distance between the two top coordinates with the number of columns the grid will include. Next, you loop from top right to top left in intervals of cell width storing the coordinates as you move, for each iteration you also store the coordinates from the interval coordinates to the bottom until the latitude value of the newest stored coordinate is lower than lowLat (to get rows). See figure below for visualisation.

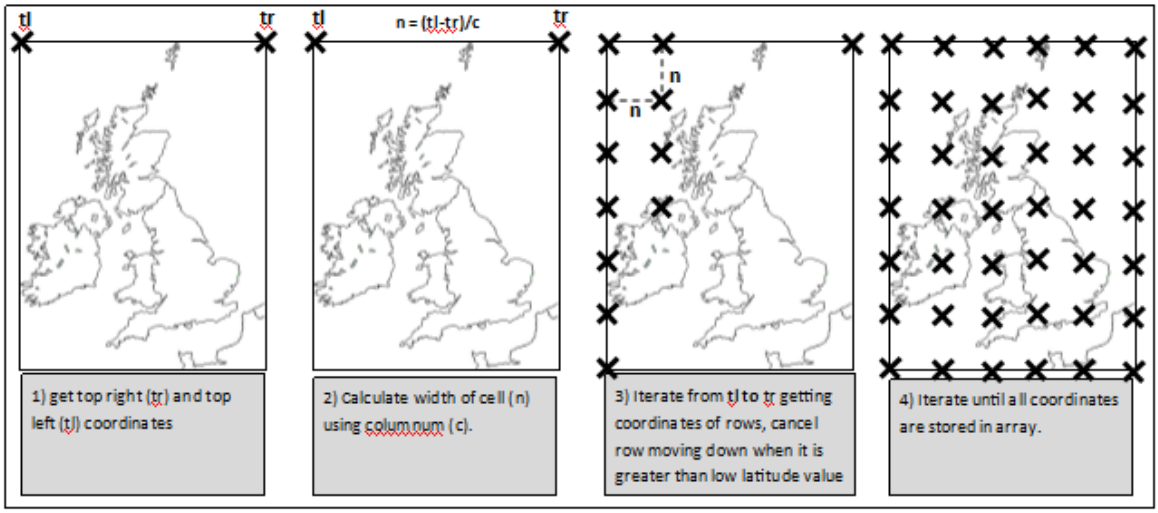


Figure 27 Design: Grid Implementation Design

As mentioned briefly above, the getCellCenterCoordinate() method is used to find the centre of a cell, this is used to display a count number or marker in the centre of a cell. The centre coordinate is calculated by finding the average of the latitude values and average of the longitude values:

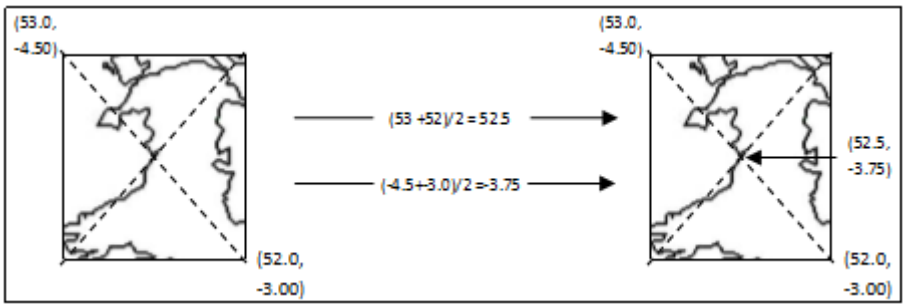


Figure 28 Design: Centre Coordinate Design

4.3.5 geoConversions

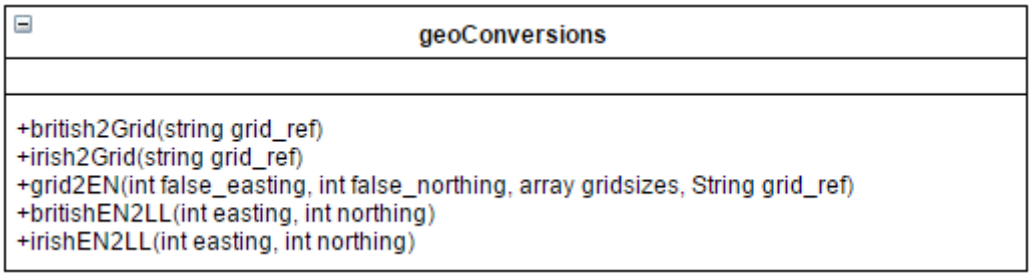


Figure 29 Design: geoConversions Class Design

british2Grid(): Converts British Grid Reference to Grid Reference
irish2Grid(): Converts Irish Grid Reference to Grid Reference
grid2EN(): Converts Grid Reference to Easting and Northing.
britishEN2LL(): Converts British Easting and Northing to Longitude and Latitude.
irishEN2LL(): Converts Irish Easting and Northing to Longitude and Latitude.

The purpose of this class is to convert any geospatial reference system to longitude and latitude. Flickr data is extracted in longitude and latitude, however NBN data is extracted as British and Irish grid reference values. These are then converted to longitude and latitude via easting and northings.

4.4 Database Design

The database will be used to simply store data during the project. While there were originally plans to perform complex geo spatial querying, this was discontinued as it was decided that the same functionality could be mimicked using Python and Python modules instead. Therefore, a simple MySQL database has been implemented which consists of separate tables with no relation to each other, one for each of the species that will be researched.

The database will be accessed using phpMyAdmin on the Cardiff University Server, allowing for a GUI view of the tables and the data. The database is accessed via many Python scripts within the project to create tables, and insert and select data. For example, createTimeline.py queries the database to collect Flickr data to be displayed in a graph, likewise dataPlot.py, mapFlickrVSNBN.py and 3DMapping.py access the database to collect Flickr data to plot on a map. The database is also queried to populate the grid for visual and similarity calculations.

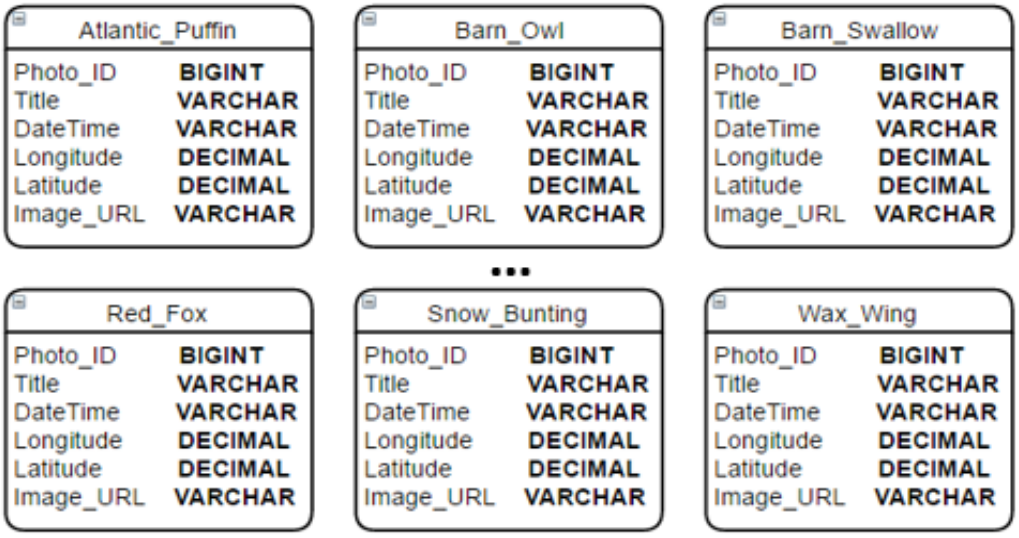


Figure 30 Design: Database Design

Photo_ID: Used to store the photo_id of a Flickr image, it is defined by Flickr not the database. The BIGINT data type has been selected as the photo id are larger than 2147483647 which is the largest value an INT can store.
Title: Used to store the title of the Flickr image, this can be used to quickly identify what the image contains. Varchar is a suitable data type as the title is simply a string.
DateTime: Stores the date the image was taken in yyyy-mm-dd hh:mm:ss format. Stored as a varchar as there won't be any functions taking place directly on the database, this data will be queried and post processed.
Longitude: Used to store the longitude coordinate value of the image. It has been stored separately to the latitude value as often the values will be searched independently.
Latitude: Used to store the latitude coordinate value of the image.
Image_URL: Used to store the URL of the Flickr image. Can be used to view the photo later if necessary. For example, to confirm that a large cluster of images taken at the same place are not duplicate images, which need to be removed.

4.5 Key Scripts

4.5.1 FlickrDataCollection.py

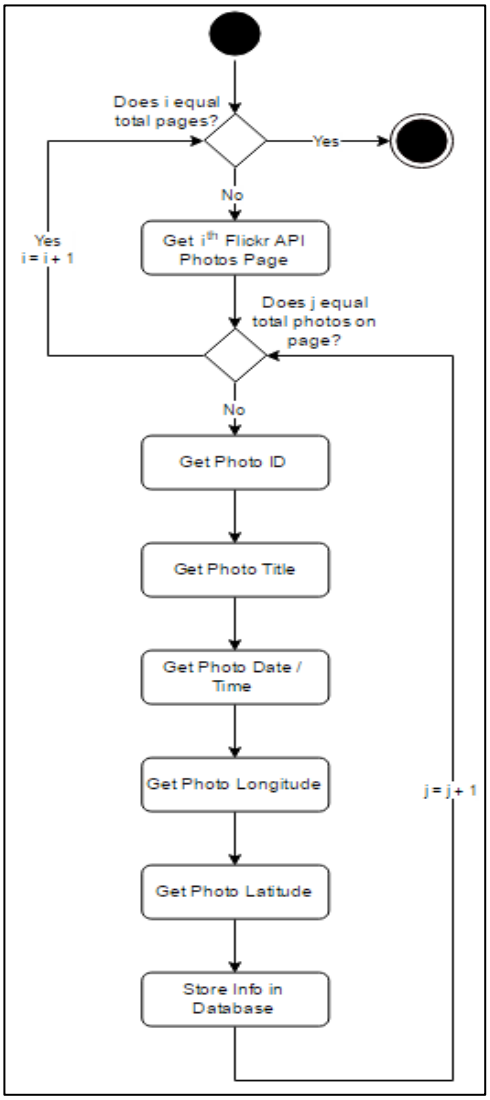


Figure 31 Design: Activity Diagram - Flickr Data Collection

This script is will be used to collect data from Flickr. It will require use of Python package FlickrAPI in order to query the API using calls such as Flickr.search. Python package xml will be needed to navigate the output of the flickAPI calls, and finally pyMySQL will be needed to store collected data.

The algorithm contains two for loops. The Flickr API return results in a series of pages, each of which contains 250 photos, therefore the first loop is used to loop through each pages, and the nestled loop is used to extract the desired data for each photo on the page.

The algorithm will require two calls to the API for each photo. The first is the generic search that will return every photo, the second is an individual search for each photo. The individual search is necessary to collect date, time, longitude, latitude, and URL data. See figure 1 in the appendix for the algorithm.

4.5.2 createTimeline.py

This script is used to create a timeline for a specified species, between a specified start date and the end of the data. This script makes use of Python package matplotlib to create either a bar chart or line graph, and pyMySQL to query the database to collect data.

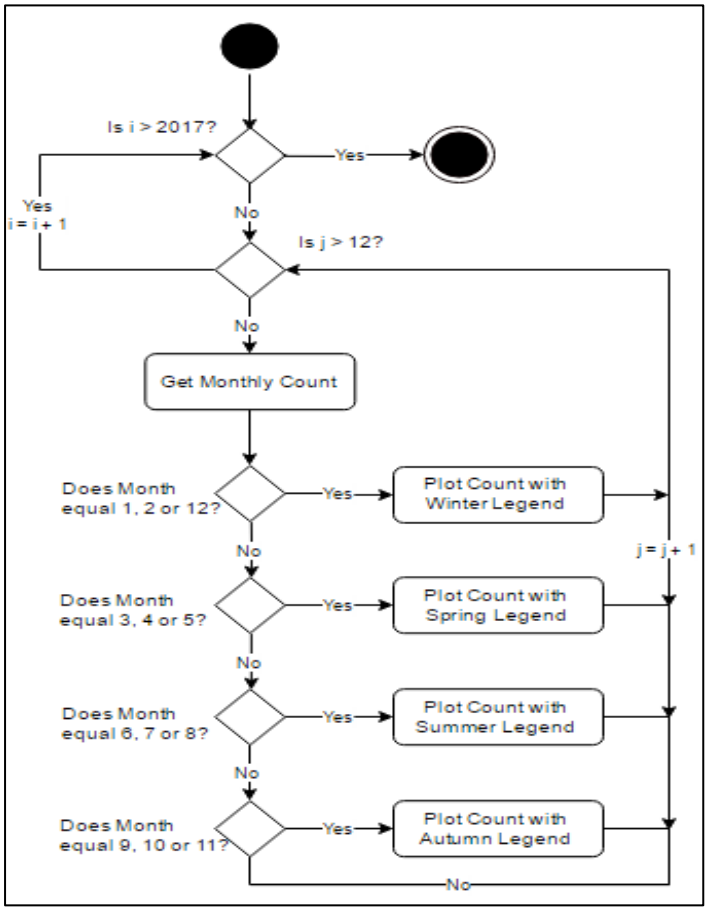


Figure 32 Design: Activity Diagram - Bar Timeline

Figure 30 displays a flow used to create a bar chart displaying monthly total captured photos, it uses a for loop to iterate through the specified years, and a nestled for loop to iterate through the months. The nestled loop contains several if statements that are used to specify the season, allowing the colour of the bars to be adjusted so to show seasonal variation more clearly. See figure 2 in the appendix for the algorithm.

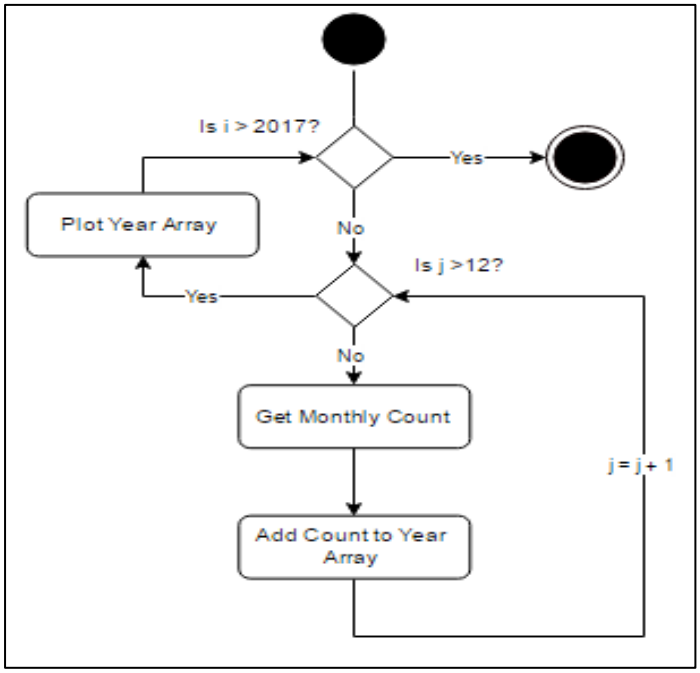


Figure 33 Design: Activity Diagram - Line Timeline

Figure 31 displays a flow for a line graph that will display a different line for each year within the specified range. Its purpose is to show years with high photo counts, and to show what time of year the species is most commonly photographed. It uses the same for loop set up to loop through each month, it then stores a years' worth of data in an array and plots it as a complete line after the nestled loop. See figure 3 in the appendix for the algorithm.

4.5.3 processNBNData.py

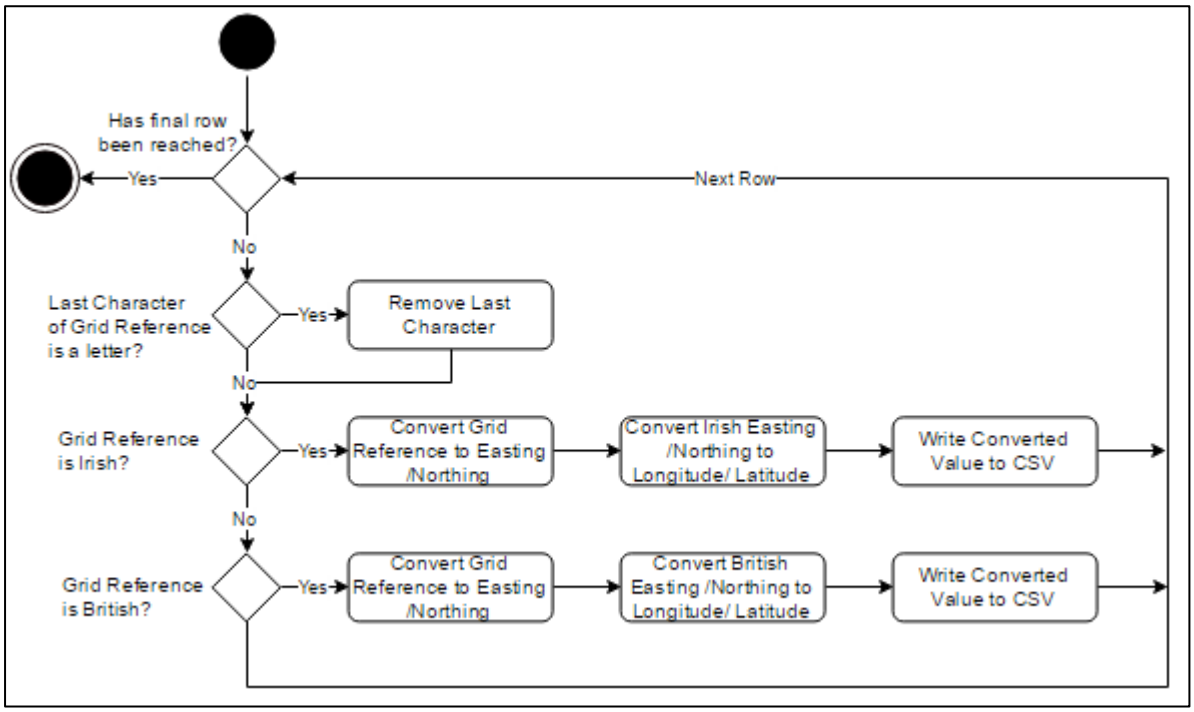


Figure 34 Design: Activity Diagram - Convert British and Irish Grid Reference to Longitude and Latitude

The flow displayed in Figure 32 is used to convert the British National Grid and Irish Grid Reference coordinate systems used by NBN into Latitude and Longitude values. The script makes use of Python package pyproj to convert British easting/northing, and Irish easting/northing to latitude and longitude. Note the last letter is removed, this is due to the conversion functions not being able to deal with reference not ending in a number. See figure 4 in the appendix for the algorithm.

The script uses a for loop to iterate though every line of the NBN CSV file. Within the loop it first collects the grid reference from the 6th column of the row in the CSV, it then uses a series of IF statements to decide if the grid reference is British or Irish, so that the correct class functions can be used for conversion.

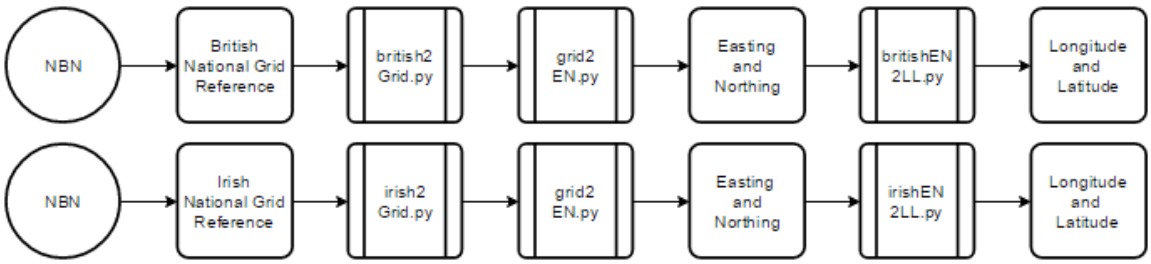


Figure 35 Design: Data conversion data flow

4.5.4 dataPlot.py

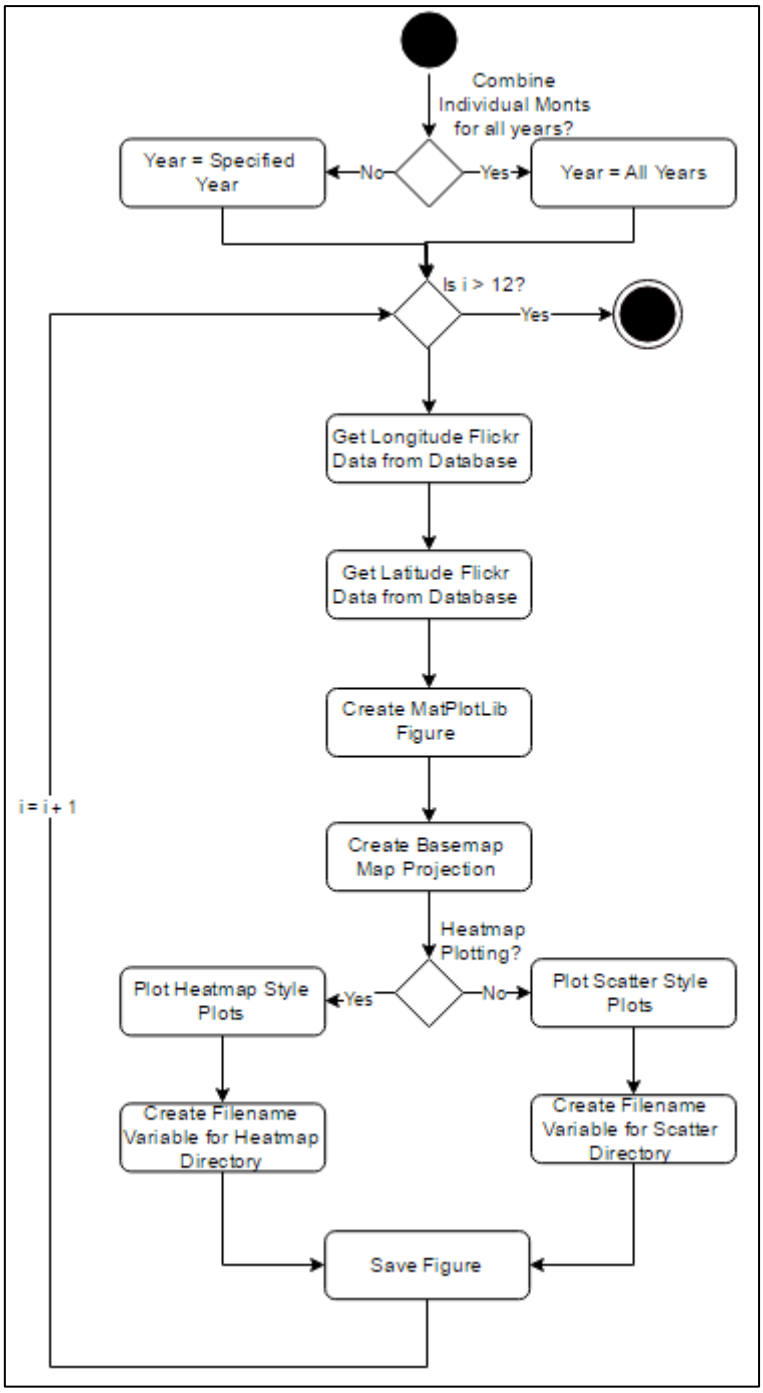


Figure 36 Design: Activity Diagram - Plotting Data

The flow displayed in Figure 34 shows how the SQL data is plotted on a map in separate time slices. A for loop is used to iterate through each month taking all the longitude and latitude values for the corresponding month either a specific year or all years depending on a user specified parameter. The coordinates are then displayed on a map projection, and saved to an appropriate directory. In total, up to 12 time slices can be saved for each use of the script.

5.0. Implementation

5.1 Flickr Data Collection

5.1.1 URL API Calls

By providing details of the project to Flickr an API key was given to be used for all API calls. Below is an example API call using a web browser:

https://API.Flickr.com/services/rest/?method=Flickr.photos.search&API_key=18ae9dcc4c7b0f369137889d23552c75&tags=redfox,vulpesvulpes&place_id=cnffEpdTUb5v258BBA%27

Each API call is split into three sections. They are the API URL, the method, and the parameters:

1. API URL: This is the same for all API calls. https://API.Flickr.com/services/rest/
2. Method: Flickr offers many methods that can be called, the full extent can be found at https://www.Flickr.com/services/API/, examples include Flickr.groups.browse, Flickr.people.findByEmail, and Flickr.photos.search.
3. Parameters: Each method has different optional and non-optional parameters that can be applied. API key is an example of a non-optional parameter. Tags and place_id are examples of optional parameters used to refine the search.

Below is an output of the URL based API call:

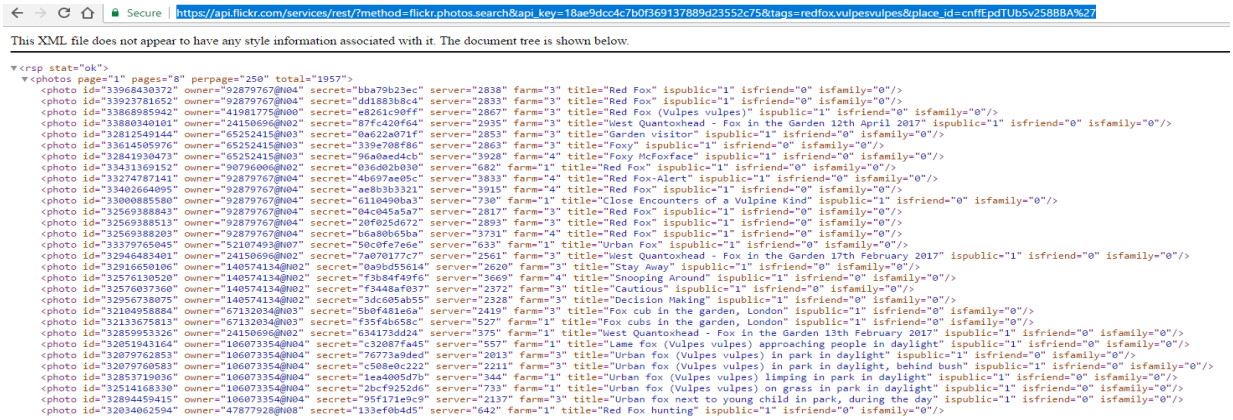


Figure 37 Implementation: Flickr API URL Query Output

5.1.2 FlickrAPI Python Module Implementation

Utilising the FlickrAPI Python module is very similar to the URL method. You first set up your Flickr connection using *Flickr = FlickrAPI.FlickrAPI(API_key, API_secret).* Then you run methods like the following:

```
photos = flickr.photos.search(tags =redfox + vulpesvulpes ,place_id='cnffEpdTUb5v258BBA' )
```

Note the method and parameters are used almost identically to the URL version. In the example above the variable 'photos' will contain an xml etree, which can then be searched using xml Python package as follows:

```
photo_id = photos[0][1].get('id')
```

This example will get the photo_id from the first row of the xml. The API is used once for each species, the data is extracted and stored in the database.

5.2 Database

5.2.1 Creating Database Connection

Utilising the pyMySQL Python package you can set up a database connection and use said connection multiple times per script. The connection is set up using the details of the database as variables in a pyMySQL connection function:

```
import pymysql
host = 'ephesus.cs.cf.ac.uk'
user = 'c1317264'
password = '********'
db = 'c1317264'
conn = pymysql.connect(host, user, password, db, charset='utf8')
cursor = conn.cursor()
```

The cursor variable is then used to execute any queries.

5.2.2 Creating Tables

Due to the simple design of the database consisting of several standalone tables, creating tables only requires a single SQL command to be executed per table.

```
CREATE TABLE Red_Fox (
Photo_ID BIGINT(6) NULL,
Title VARCHAR(500) NULL,
Date_Time VARCHAR(500) NULL,
Longitude DECIMAL(10,6) NULL,
Latitude DECIMAL(10,6) NULL,
Image_URL VARCHAR(500) NULL)
```

This query will create a table named 'Red_Fox' and will add the columns and data types as designed previously. This is a script was useful to save as a template as throughout the iterations of the project many tables have been added for species not previously planned to be researched.

5.2.3 Inserting Data into Tables

Adding data to the database is done infrequently due to it only needing to be done once for each species. Below is an example of inserting data into a table and using cursor to execute the query.

```
query = "INSERT INTO Red_Fox (Photo_ID, Title, Date_Time, Longitude, Latitude, Image_URL) VALUES ("+
        photo_id +","+ photo_title +","+ datetime +","+ longitude +", "+ latitude +","+ url+")"

cursor.execute(query)
```

5.2.4 Selecting Data

Using select queries is a large part of many of the scripts. All plotting, timeline, similarity calculations, and visual comparison scripts use select queries to extract the necessary coordinate data.

```
query = "SELECT "+columnName+" FROM "+tableName+" WHERE YEAR(Date_Time)>="
+minYear+" AND Year(Date_Time)<="+maxYear+" AND MONTH(Date_Time)>="
+minMonth+" AND MONTH(Date_Time)<="+maxMonth

cursor.execute(query)
```

Here is an example query used to extract user specified data between user specified dates. This script is used as part of method getSQLData() from the class entitled plotting.

5.3 Plotting Maps

5.3.1 Creating A Map

The first step for plotting data on a map is to create a figure to plot the data on. Next, you create the map that will be projected in front of the matplotlib figure. As previously mentioned in the methods section a Python package called Basemap is used to create all the maps, furthermore in the design section it is briefly explained that there is a class called mapping which includes a method called createMap. This method is the first step towards plotting any data.

```
def createMap(self, lowLat, highLat, leftLon, rightLon, shapefile='no', river='no'):

    m = Basemap(projection='mill',llcrnrlat=lowLat,urcrnrlat=highLat,\
    llcrnrlon=leftLon,urcrnrlon=rightLon,resolution='h')
    m.drawcoastlines(zorder=3)
    m.fillcontinents(color='white',lake_color='grey',zorder=0)
    m.drawmapboundary(fill_color='white',zorder=0)

    if shapefile == 'yes':
        m.readshapefile('Shapefiles/Areas/Areas', 'areas',zorder=3)

    if river == 'yes':
        m.drawrivers(linewidth=0.5, linestyle='dashed', color='blue', antialiased=1, ax=None, zorder=3)

    return m
```

Basemap(): Used to create the map projection. The parameters for this are projection which allows you to select projection type (mill stands for Miller Cylindrical Projection), coordinate parameters which define the area the projection will show, and finally the resolution defines the quality of the graph (h stands for high, c for crude is also useful for displaying data quickly a map with poor resolution).
m.drawcoastlines(): Draw a coastline on the map projection. Can specify weight of coastline.
m.fillcontinents(): Colour the land areas on map projection. Can specify land and lake colour.
m.drawboundary(): Define the colour for the rest of the map.
m.readshapefile(): Display a shapefile on the map projection. The 'Areas' example above shows all counties with in the UK.
m.drawrivers(): Draw all major UK rivers on the map projection.

5.3.2 Collecting Data to Plot

There are two sources of data to plot, SQL and CSV, both of which require different methods to extract data from. SQL uses queries as described in section 5.2.4 to collect data for a user defined species and time range. CSV data involves using the CSV Python package to open the CSV file, then use a for loop to iterate though each line of the CSV collecting the necessary data. This is done using method getCSVData() from the class entitled plotting.

```
def getCSVData(self, filepath):
    lonArray = []
    latArray = []
    csvSource = open(filepath,'rb')
    reader = csv.reader(csvSource)

    iterreader = iter(reader)
    next(iterreader)
    for row in iterreader:
        lonArray.append(float(row[0]))
        latArray.append(float(row[1]))
    return(lonArray, latArray)
```

The latitude and longitude values are collected separately, this makes the step of plotting the data on the figure much easier. The iter tool is also used to skip the title line of the CSV file.

5.3.3 Plotting Data

Once the map is created and the data is collected, plotting the data requires few lines. The unique function required is a Basemap function that converts longitude and latitude values from coordinates to matplotlib markers that can be plotted on the figure behind the map projection, see the function below (lon and lat are arrays):

```
x, y = m(lon, lat)
```

```
[-0.195743, -0.092267, -1.28746, -1.105971, -1.105971, -1.105971, -1.105971, -1.105971, -1.650974, -0.550684, -0.3501
46, -2.215869, -0.610985, -1.328015, -0.276954, 0.105657, -0.707073, -0.257492, -0.272941, -0.053644, -0.204931, 0.65
5231, 1.060137, 1.060137, -0.029439]
```

Latitude values before conversion.

```
[1256975.435996727, 1268481.436808209, 1135582.101427278, 1155762.7479663305, 1155762.7479663305, 1155762.7479663305,
 1155762.7479663305, 1155762.7479663305, 1095161.2078965565, 1217507.816123248, 1239806.6138225496, 1032347.779387511
4, 1210802.6540090076, 1131072.5933006627, 1247945.189061194, 1290489.5711061303, 1200118.1609287413, 1250109.2637045
225, 1248391.4140916984, 1272776.1164377062, 1255953.7774917996, 1351599.3829462596, 1396622.8547133727, 1396622.8547
133727, 1275467.588369767]
```

Latitude values after conversion to plottable values.

The next step is to run the matplotlib plotting function and display the figure.

```
plt.scatter(x, y, s=10, c='r',zorder=4)
plt.show()
```

5.4 Heatmaps

Simple heatmaps may be implemented using matplotlib and basemap, similarly to plotting a standard scatter plot. The main differences are that the below variables must be defined to create a heatmap plot:

```
heatmap, xedges, yedges = np.histogram2d(x, y, bins=40)
extent = [xedges[1], xedges[-2], yedges[1], yedges[-2]]
```

Then the x,y coordinates are plotted using the following line:

```
plt.imshow(heatmap.T, extent=extent,filterrad=10,alpha=0.8,vmin=1, vmax=20,
                                                    interpolation='sinc' ,origin='lower', zorder=2,  cmap='Blues')
```

5.5 3D Mapping

Creating 3D maps is very similar in practice to 2D mapping. It follows similar steps:

1. Create a matplotlib figure
2. Create a matplotlib Axes3D object
3. Create a Basemap map projection. To draw coastlines and countries you need to encase the functions detailed in section 5.3.1 with ax.add_collection3d(m.drawcoastlines())
4. Collect the data from either CSV or SQL storage.
5. Convert data from longitude and latitude to plottable data using Basemap function.
6. Run the following function, to get the preceding output:

```
ax.bar3d(x, y, z, dx, dy, dz, color= 'r', alpha=0.8)
```

x: The x coordinate, the same as a 2D map.
y: The y coordinate, the same as a 2D map.
z: The z coordinate for plotting, must be set to 0 for Basemap projection.
dx: Width of data bar.
dy: Length of data bar.
dz: Height of data bar.

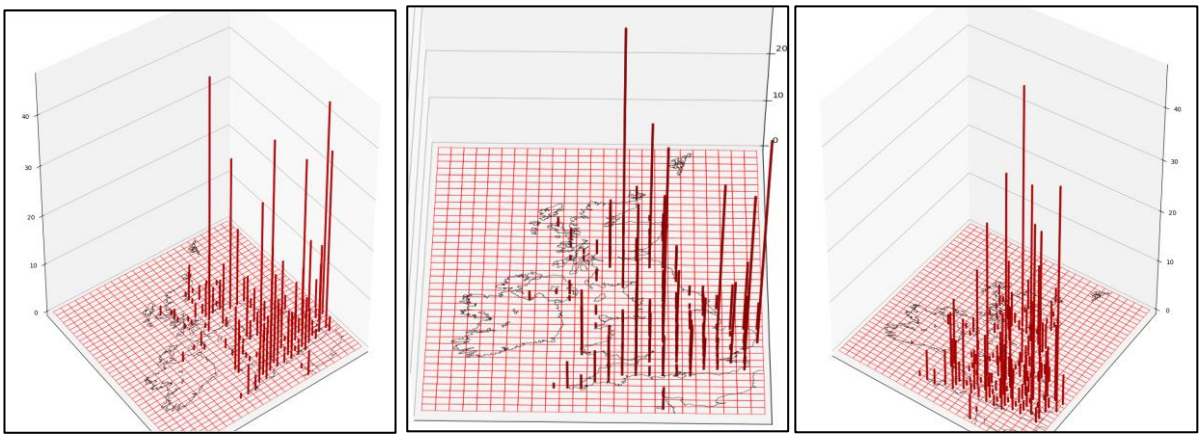


Figure 38 Implementation: 3D Mapping Example

5.6 Timeline

Creating a timeline is done using Python package matplotlib. They can be implemented in various formats, such as bar charts, line graphs, and scatter graphs. The most identifiable feature of a timeline is date values on the x axis of the figure. The timelines were created using the following steps:

1. Create a matplotlib figure.
2. Collect the data from CSV or SQL.
3. Plot the data on the graph using plt.bar for bar charts, or plt.plot for line graphs.
4. Use plt.show() or plt.savefig() to output the timelines.

Optional formatting can be completed too, for example:

X and Y Axis Labels	`plt.xlabel('Date', fontsize = 10)` `plt.ylabel('Total Photos Taken', fontsize = 10)`
Legend	`plt.legend()`. This requires you to label your plots: `plt.plot(x,y, label=year)`
Figure Title	`plt.title('Red_Fox : 2000 - 2017', fontsize = 15)`

5.7 Grid

The grid is useful research tool in this project. It facilitates the ability to view areas with many captured photographs that cannot be viewed using plots alone, it provides an opportunity to more accurately compare NBN and Flickr data via visualisation, and enables the similarity calculations of the two datasets to be completed. In section 4.3.4 the design of the grid implementation is explained, the design is then implemented in method createGrid().

```
def createGrid(self, columnNum, lowLat, highLat, leftLon, rightLon):
    lonArray = []
    latArray = []
    squareWidth = ((leftLon - rightLon)/columnNum)
    plotLon = leftLon
    tempRowNum = 0
    while plotLon <= rightLon:
        #plot the top of each column
        lonArray.append(plotLon)
        latArray.append(highLat)
        tempRowNum= tempRowNum+1
        plotLat = highLat
        #for each column plot each of the rows.
        while plotLat >= lowLat:
            plotLat = plotLat + (squareWidth/1.75)
            lonArray.append(plotLon)
            latArray.append(plotLat)
            tempRowNum= tempRowNum+1
        plotLon = plotLon - squareWidth
        rowNum = tempRowNum
        tempRowNum = 0
    lowLat = plotLat
    return(lonArray,latArray,lowLat,rowNum)
```

As designed the algorithm iterates through the columns of the grid, while a nestled loop iterates through the rows. The coordinates of each corner of the cells are stored in an array starting with the top left cell and going down until the bottom right coordinate is stored.

Due to the nature of the outputted arrays it is simple to then search the grid. A method called getCellByID() is used in conjunction with the grid in all relevant scrips, and is used to get the four corners of the cell. This is achieved by the top left corner which is equal to the ID parameter, the bottom left is ID + 1, the top right is ID + total number of rows, and the bottom right is ID + total numbers of rows + 1.

```
getCellLat = []
getCellLat.append(gridLatArray[squareID])
getCellLat.append(gridLatArray[squareID+1])
getCellLat.append(gridLatArray[squareID+rowNum+1])
getCellLat.append(gridLatArray[squareID+rowNum])
```

Using the cell coordinate you can then use a SQL query, or CSV search, to locate all the geotags that appear within the cell. If the geotags longitude coordinate is >= to the cells top left coordinate, and <= to the bottom right coordinate, and the latitude is <= to the top left coordinate, and >= to the top right coordinate then the geotag is within the cell. See below for an example searching the CSV:

```
next(iterreader)
for row in iterreader:
    if float(row[0]) >= getCellLon[0] and float(row[0]) <= getCellLon[3]
    and float(row[1]) <= getCellLat[0] and float(row[1]) >= getCellLat[1]:
        count = count + 1
return(count)
```

The singular cell coordinates can also be used to find the centre of a cell, using algorithm defined in section 4.3.4.

5.8 Evaluation Techniques

5.8.1 Generating Data Sets for Comparison

The grid is used to create the dataset for each of the similarity calculation. A grid is created with 25 columns and 39 rows and created an array of the total geotags for a specified species in each cell that covers land with the UK, and disregarded all cells covering sea or Ireland. See below for grid cells that were used for comparison:

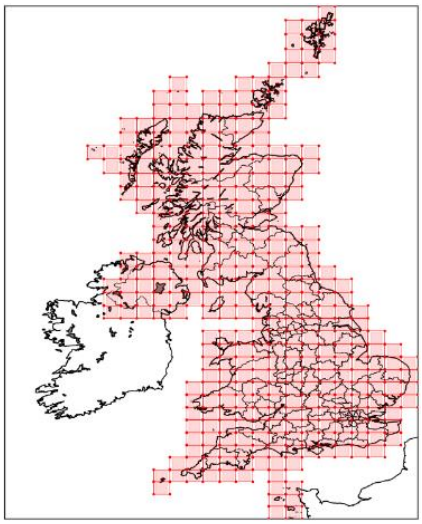


Figure 39 Implementation: Grid Search Space

The sea and Ireland cells are disregarded due to them containing zero data, therefore they would add noise to the calculations. There are 635 disregarded cells in total, each of which would have added an additional 0 value to the array.

The array of cell counts was created for Flickr and NBN data, each array has equal length and each key corresponds to the same cell. Prior to being used for calculations the data is first normalised due to the NBN dataset being much larger. The normalisation process first converts the counts into a percentage between 0 and 100, percentage values are then rounded to the closest 0.5% (therefore all areas of NBN and Flickr data that contain very few results are normalised to 0).

5.8.2 Calculations

Hellinger Distance: The data set is first converted from a percentage between 0-100 to a percentage between 0-1. The Hellinger algorithm is then implemented in Python using Python package numpy.

```
np.sqrt(np.sum((np.sqrt(csvPD) - np.sqrt(sqlPD)) ** 2)) / _SQRT2
```

*Algorithm taken from https://gist.github.com/larsmans/3116927 [43]

Confusion Matrix: The confusion matrix is implemented using a for loop to iterate through each of the values in each dataset, then a series of if statements to determine if the data has a true positive, true negative, false positive or false negative relationship. The total of each relationship can then be printed in a matrix format, or returned for use in further calculations.

```
for i in range(0,total):

    if (csvCountArray[i] != 0 and sqlCountArray[i] != 0):
        truePositive = truePositive + 1

    if (csvCountArray[i] == 0 and sqlCountArray[i] ==0):
        trueNegative = trueNegative + 1

    if (csvCountArray[i] != 0 and sqlCountArray[i] ==0):
        falseNegative = falseNegative + 1

    if (csvCountArray[i] == 0 and sqlCountArray[i] !=0):
        falsePositive = falsePositive + 1
```

*Implemented following mathematical method on http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/ [44]

Precision: Precision is implemented using the true positive and false positive values from the confusion matrix.

```
float(truePositive)/(falsePositive+truePositive)
```

Recall: Recall is implemented using true positive and false negative values from the confusion matrix.

```
float(truePositive)/(truePositive+falseNegative)
```

Accuracy: Accuracy is implemented using true positive, true negative values from the confusion matrix, and the total values in the dataset.

```
(truePositive+trueNegative)/float(total)
```

F1 Score: F1 score is implemented using precision and recall values.

```
2 * ((precision * recall) / (precision + recall))
```

R-Squared: R Squared requires you to calculate the mean of each data set, each value in each array minus the mean, each value squared, each value multiplied by its equivalent value in the adjacent dataset, and the sum of each of these calculated values. See below for the algorithm:

```
xmean = sum(sqlCountArray)/len(sqlCountArray)
ymean = sum(csvCountArray)/len(csvCountArray)

for i in range(0,len(sqlCountArray)):

    a = sqlCountArray[i]-xmean
    b = csvCountArray[i]-ymean

    ab = (a * b)
    abArray.append(ab)

    a2 = (a * a)
    a2Array.append(a2)

    b2 = (b * b)
    b2Array.append(b2)

n = len(sqlCountArray)
abSum = sum(abArray)
a2Sum = sum(a2Array)
b2Sum = sum(b2Array)

correlationCoefficient = abSum / math.sqrt(a2Sum*b2Sum)
determinationcoefficient = correlationCoefficient*correlationCoefficient
```

*Implemented following mathematical method on https://www.mathsisfun.com/data/correlation.html [45]

5.9 Geo Conversion Methods:

5.9.1 Irish and British National Grid to Easting and Northing

To convert Irish and British National Grid to Easting and Northing a Python scripts from snorfalorpagus.net were utilised. [3] The scripts were edited slightly to work with the projects classes and scripts. The first two figures are used to define the false easting and northing values, and the gridsize for the British or Irish national Grids. The third script converts the grid reference to easting and northing.

```
def british2Grid(self, grid_ref):
    false_easting = 1000000
    false_northing = 500000
    gridsizes = [500000, 100000]
    return(false_easting, false_northing, gridsizes, grid_ref)
```

```
def irish2Grid(self, grid_ref):
    false_easting = 0
    false_northing = 0
    gridsizes = [100000]
    return(false_easting, false_northing, gridsizes, grid_ref)
```

```
def grid2EN(self, false_easting, false_northing, gridsizes, grid_ref):
    # false easting and northing
    easting = -false_easting
    northing = -false_northing
    alphabet = 'ABCDEFGHJKLMNOPQRSTUVWXYZ'

    # convert letter(s) to easting and northing offset
    for n in range(0, len(gridsizes)):
        letter = grid_ref[n]
        idx = alphabet.index(letter)
        col = (idx % 5)
        row = 4 - ((idx) / 5)
        easting += (col * gridsizes[n])
        northing += (row * gridsizes[n])

    # numeric components of grid reference
    grid_ref = grid_ref[len(gridsizes):] # remove the letters
    e = '{:0<5}'.format(grid_ref[0:len(grid_ref)/2])
    e = '{}.{}'.format(e[0:5],e[5:])
    n = '{:0<5}'.format(grid_ref[len(grid_ref)/2:])
    n = '{}.{}'.format(n[0:5],n[5:])
    easting += float(e)
    northing += float(n)

    return easting, northing
```

5.9.2 Easting and Northing to Longitude and Latitude

To convert the Easting and Northing values to Longitude and Latitude a script from webscraping.com [52] was utilised. The script was edited slightly to allow for Irish Easting and Northing to be converted also. The original value vgrid = Proj(init="world:bng") is used for British National Grid only (bng), it was converted from world:bng to epsg:2270 for British grid and epsg:29903 for Irish grid.

```
def britishEN2LL(self, easting, northing):
    v84 = Proj(proj="latlong",towgs84="0,0,0",ellps="WGS84")
    v36 = Proj(proj="latlong", k=0.9996012717, ellps="airy",
            towgs84="446.448,-125.157,542.060,0.1502,0.2470,0.8421,-20.4894")
    vgrid = Proj(init="epsg:27700")

    vlon36, vlat36 = vgrid(easting, northing, inverse=True)
    return transform(v36, v84, vlon36, vlat36)
```

```
def irishEN2LL(self, easting, northing):
    v84 = Proj(proj="latlong",towgs84="0,0,0",ellps="WGS84")
    v36 = Proj(proj="latlong", k=0.9996012717, ellps="airy",
            towgs84="446.448,-125.157,542.060,0.1502,0.2470,0.8421,-20.4894")

    vgrid = Proj(init="epsg:29903")
    vlon36, vlat36 = vgrid(easting, northing, inverse=True)
    return transform(v36, v84, vlon36, vlat36)
```

6.0 Testing

6.1 Accuracy of Evaluation Implementations

6.1.1 R Squared Testing

To test the accuracy of the R Squared calculation the two datasets are extracted and compared. The Flickr and NBN dataset for the Atlantic Puffin was used:

```
Flickr Dataset:
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0
.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 10.5, 0.0, 0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 13.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0
.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.5, 0.5, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 15.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0
.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 6.5, 3.0, 0.0, 0.0, 0.0,
32.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 2.5, 0.0, 0.5, 0.0, 0.0, 0.0, 0
.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 5.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 0.0, 1.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]

NBN Dataset:
[0.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0
.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.5, 0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0, 2.5
, 3.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 15.5, 0.0, 4.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 4.0, 1.0, 0.0, 1.0, 0
.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 2.5, 0.0, 0.5, 0.0, 0.0, 0.0, 1.5, 0.0, 0.0
, 2.5, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.5, 0.0, 0.0,
0.0, 0.0, 0.0, 1.5, 0.0, 0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.5, 0.0, 2.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0
.0, 0.5, 0.0, 0.5, 0.0, 0.0, 0.5, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5, 3.5, 0.5, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.5, 5.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5, 1.0, 0.5, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0
.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 4.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 15.5, 0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 1.0, 2.0, 2.5, 0.0]
```

The datasets are pasted into an Excel document and R Squared is calculated following two methods. The first was to use the built in excel function called CORREL to calculate the coefficient of correlation (r), then squared the results to get R^2. The second method was to calculate the value following the exact steps that the Python function follows:

x	y	a	b	a*b	a2	b2
0.00	0.5	-0.27	0.24	-0.063284929	0.071747	0.05582
0.00	0	-0.27	-0.263736264	0.070643642	0.071747	0.06956
0.00	0	-0.27	-0.263736264	0.070643642	0.071747	0.06956
0.00	0	-0.27	-0.263736264	0.070643642	0.071747	0.06956
0.00	0	-0.27	-0.263736264	0.070643642	0.071747	0.06956
0.00	0	-0.27	-0.263736264	0.070643642	0.071747	0.06956
0.00	0.5	-0.27	0.236263736	-0.063284929	0.071747	0.05582
0.00	0	-0.27	-0.263736264	0.070643642	0.071747	0.06956
0.00	0	-0.27	-0.263736264	0.070643642	0.071747	0.06956
0.00	0	-0.27	-0.263736264	0.070643642	0.071747	0.06956
0.00	0	-0.27	-0.263736264	0.070643642	0.071747	0.06956
0.50	1	0.23	0.736263736	0.170918367	0.05389	0.54208

Excel Formula	
CORREL(L82:L445,M82:M445)	-0.01360888
(O68*O68)	0.000185202

Manual Calculation	
x Mean:	0.267857143
y Mean:	0.263736264
a*b Sum:	-13.7142857
a2 Sum:	1644.133929
b2 Sum:	617.6813187
R:	-0.01360888
R Squared:	0.000185202

The excel results are then compared with the output of the R Squared Python function to check the values matched.

```
X Mean:     0.267857142857
Y Mean:     0.263736263736
a*b Sum:    -13.7142857143
a2 Sum:     1644.13392857
b2 Sum:     617.681318681
R:          -0.0136088793525
R Squared:  0.000185201597231
```

You can see that the value for R Squared is 0.000185202 in all three cases.

6.1.2 Confusion Matrix

To test the accuracy of the confusion matrix function the same Atlantic Puffin dataset was extracted as used in the R Squared testing. Once again, the data was pasted into excel and used an excel formula to calculate the values for True Positive, True Negative, False Positive, and False Negative:

TruePositive:	7
TrueNegative:	303
FalsePositive:	11
FalseNegative:	43

x	y
0.00	0.5
0.00	0
0.00	0
0.00	0
0.00	0
0.00	0
0.00	0.5
0.00	0
0.00	0
0.00	0
0.00	0
0.50	1

The following formulas were used to calculated each values:

- True Positive: COUNTIFS($L$82:$L$445,"<>0",$M$82:$M$445,"<>0")
- True Negative: COUNTIFS($L$82:$L$445,0,$M$82:$M$445,0)
- False Positive: COUNTIFS($L$82:$L$445,"<>0",$M$82:$M$445,0)
- False Negative: COUNTIFS($L$82:$L$445,0,$M$82:$M$445,"<>0")

The range L82:L445 is the Flickr data, and the range M82:M445 is the NBN data. The formulas count the cells in the range if they fit the criteria. For example, true positive must both not equals zero (<>0)

The next step in the test was to calculate a confusion matrix for the same datasets using the Python function. The results of the test are:

```
True Positive:  7
True Negative:  303
False Positive: 11
False Negative: 43
```

Both the results are the same, hence proving that the confusion matrix calculation works.

Finally, the results of the precision, recall, accuracy, and F1 score functions can also tested as they all utilise the confusion matrix in their calculations. The values were each calculated using excel, an online calculator [46], and using the Python script to ensure they were all identical:

Precison:	0.388888888889
Recall:	0.14
Accuracy:	0.851648351648
F1 Score:	0.205882352941

Sensitivity	0.1400
Precision	0.3889
Accuracy	0.8516
F1 Score	0.2059

```
Precison: 0.388888888889
Recall:   0.14
Accuracy: 0.851648351648
F1 Score: 0.205882352941
```

6.2 Accuracy of Coordinate System Conversions

To test the accuracy of the conversion of the NBN data from UK and Irish Grid Reference to Longitude and Latitude a test dataset and script were created. The test data set contains 100 unique British grid references and their longitude and latitude equivalent, which were converted using an online conversion website [48]. The script then converts each of the British grid refences to longitude and latitude using the functions and compares the output with the true longitude and latitude values.

```
import mappingClasses
import csv
convert = mappingClasses.geoConversions()

csvSource = open('britishCoordinateTest.csv','rb')
reader = csv.reader(csvSource)
iterreader = iter(reader)

next(iterreader)

match = 0
decimalPlaces = 4


for row in iterreader:
    gridref = str(row[0])

    false_easting, false_northing, gridsizes, grid_ref = convert.british2Grid(gridref)
    easting, northing = convert.grid2EN(false_easting, false_northing, gridsizes, grid_ref)
    lonlat = convert.britishEN2LL(easting, northing)

    calcualtedLat= round(float(lonlat[1]),decimalPlaces)
    calcualtedLon=round(float(lonlat[0]),decimalPlaces)

    actualLat = round(float(row[1]),decimalPlaces)
    actualLon = round(float(row[2]),decimalPlaces)

    if calcualtedLon == actualLon and calcualtedLat == actualLat:
        match=match+1

print(match)
```

The results of the test stated that the longitude and latitude values were 100% the same to 3 decimal places, and 99% to 4 decimals places.

The same test has been completed with an Irish coordinate dataset and the values were 97% the same to 3 decimal places, and 59% the same to 4 decimal places.

3 decimal degrees are accurate within 110m, and 4 decimal degrees is accurate within 11m therefore these values for British and Irish conversion are accurate enough for their intended use. [47]

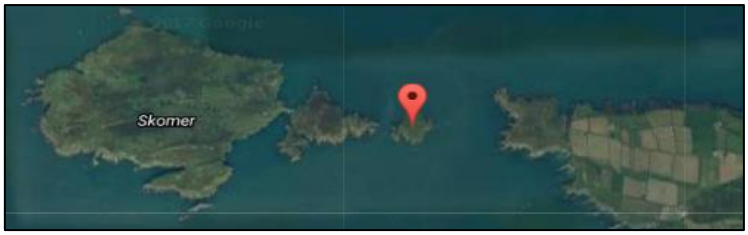
See below for the format of the test datasets in CSV format.

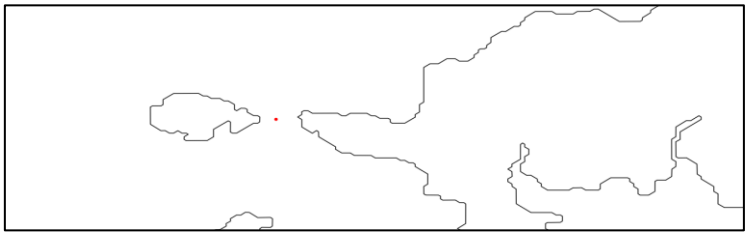
Grid Reference	Latitude	Lonitude
C3907	54.909132	-7.392711
C407041	54.882949	-7.366618
C421126	54.959180	-7.343564
C5022	55.042880	-7.218616
C5214	54.970822	-7.188786

6.3 Accuracy of Basemap Plotting

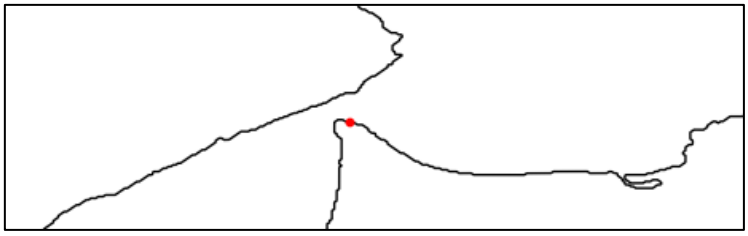
To test the accuracy of the Basemap plotting function a series of locations within the UK were selected to plot [2]. Areas were selected that were easily identifiable landforms to aid in visually accessing the quality of plot.

Skomer Island, Wales: 51.735045, -5.264425

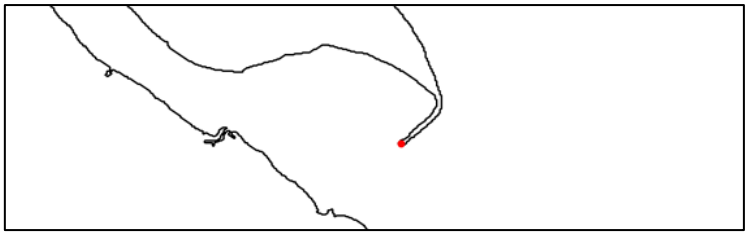




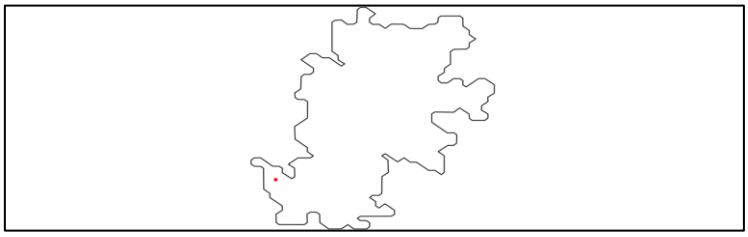
Magilligan Point, Northern Ireland: 55.193678, -6.958466



Spurn Point, England: 53.573307, 0.109863



Fair Isle, Scotland: 59.521962, -1.656060



Based on a visual comparison of the above images the accuracy of Basemap plotting function is acceptable for the requirements of the project.

The following test script was created to plot these points on a UK map, matplotlib then supplied a zoom tool which was utilised to narrow down on the plotted points for comparison.

```
lon = [-5.264425, -1.656060, 0.109863, -6.958466]
lat = [51.735045, 59.521962, 53.573307, 55.193678]

mapfunc = mappingClasses.mapFunctions()
plotfunc = mappingClasses.plottingFunctions()

plt.figure(figsize=(27.5,17.5))
m = mapfunc.createMap(mapfunc.lowLat, mapfunc.highLat, mapfunc.leftLon, mapfunc.rightLon)

x, y = m(lon, lat)

plt.scatter(x, y, s=10, c='r',zorder=4)
plt.show()
```

7.0 Results

See appendix entitled 'Figures' submitted with this final report to see a full directory of figures resulting from this project in full size.

7.1 Timelines

Timelines are a useful visualisation tool for displaying a datasets fluctuation over time. They can be utilised for displaying migration and hibernation trends, as well as showing valuable data trends in Flickr data that couldn't be seen easily without these visualisations.

7.1.1 Flickr Geotagged Data

An aim of this project was to find a method for tracking and observing wildlife without physically approaching them in the wild using methods such as tagging. Therefore, it was necessary to ensure that any methods researched are scalable for future use, this is dependent on the continuing public use of social media platforms such as Flickr to share images and location details.

A script was created to extract the total number of geotagged images available to a holder of a public API key within the UK. The parameters for this API call were a place ID and a monthly date range from 2000 - 2016.

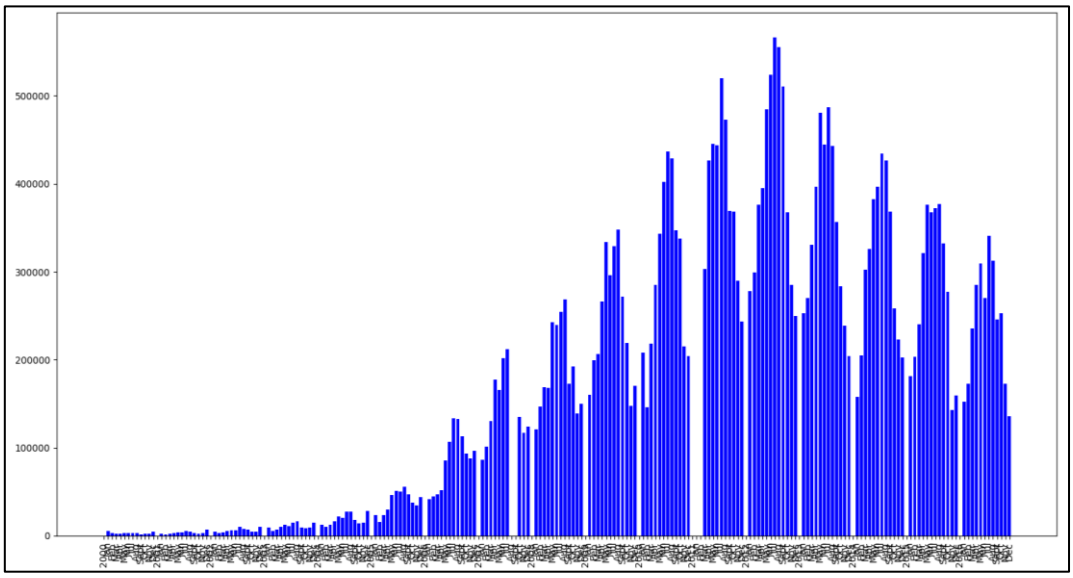


Figure 40 Results: 2000 - 2017 Total UK Geotags

The graph clearly shows a sharp rise in Flickr's popularity, starting steadily in 2006 and rapidly increasing to a peak in 2012 with a high of roughly 560,000 geotagged images. However, as of 2012 there is also an evident decline in the number of geotagged images on Flickr, which suggests that the platforms popularity is reducing. While there is still a valuable amount of data, with peaks of almost 400,000 images in a month, if the current trend continues Flickr's use as a research tool may become obsolete (especially if the poor performance leads to discontinuation of the platform).

To further understand the data, it was normalised year by year by storing the yearly total photos and dividing each month by said total to get a percentage to be plotted. This removed all spikes where years such as 2012 have considerably more data. See Figure 41.

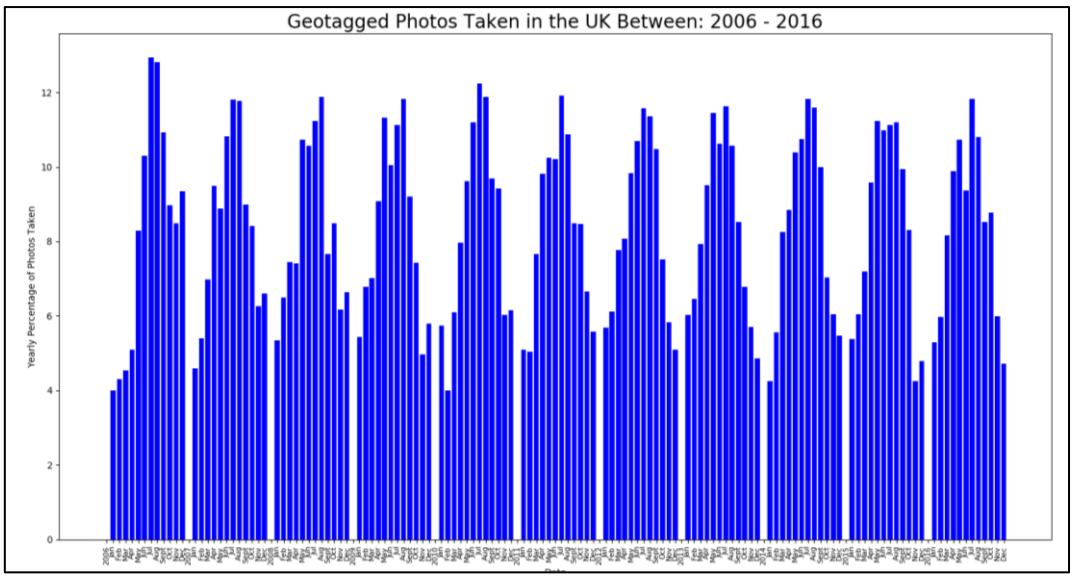


Figure 41 Results: 2006 -2016 UK Geotags, Normalised

This normalised graph clearly demonstrates a trend that summer months yield higher numbers of captured photographs, whereas winter months have considerably less content. The reason for this is not known, however it is presumed highly likely that the weather plays a significant part. As this data is generated by the public, it is likely that if it is cold or raining there will be less photos being taken. Also, in the summer school children will add to the count and working individuals will be more likely to take time away from work for holidays. Based on personal experience, people are also more likely to take large batches of photos on holiday as it part of the culture, whereas between work and school people are less likely to take large amounts photos in a short period.

Finally, the same graph was created using the tag 'wildlife' to assess if the trend was similar for wildlife photography.

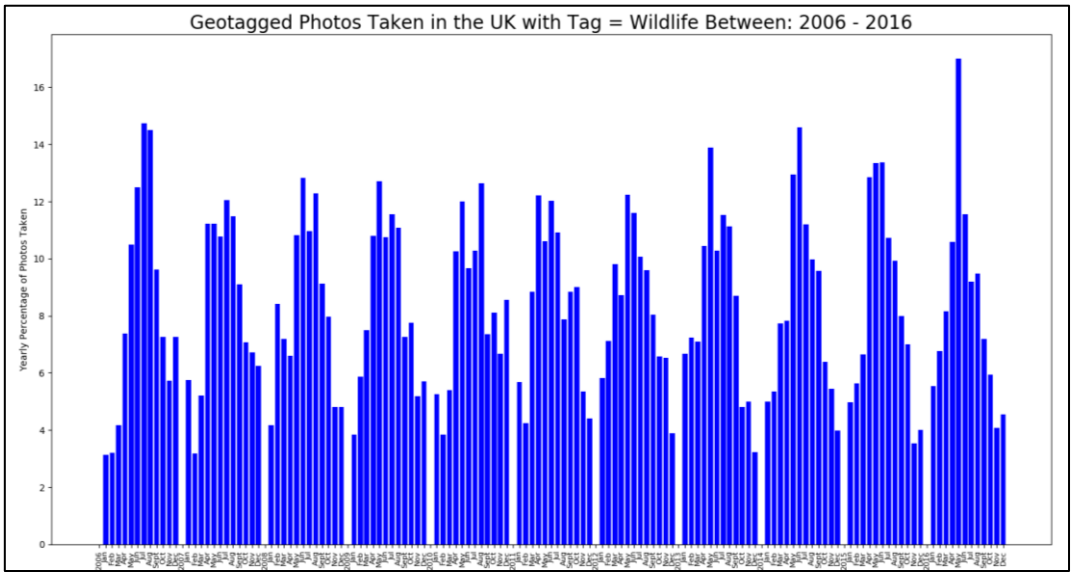


Figure 42 Results: 2006 -2016 UK Wildlife Geotags, Normalised,

It is evident that the trend is very similar, the summer months are the most popular for capturing images of wildlife in the UK. This correlates with the initial research, as UK species will be most active in the summer, and hibernating in the winter months (for example, Common Frog and Grass Snake).

This trend also suggests that the total photo count for winter species is going to be poor in comparison with summer species (for example the Snow Bunting, Brambling, and Wax Wing) proving earlier speculations that some species will be better represented by Flickr data than others.

7.1.2 Seasonal Bar Charts

Another aim of the project was to investigate if currently known wildlife trends could be confirmed using Flickr data. This bar chart has been established to help show known trends, such as bird migrations and plants blooming, and provide evidence that Flickr is a useful tool.

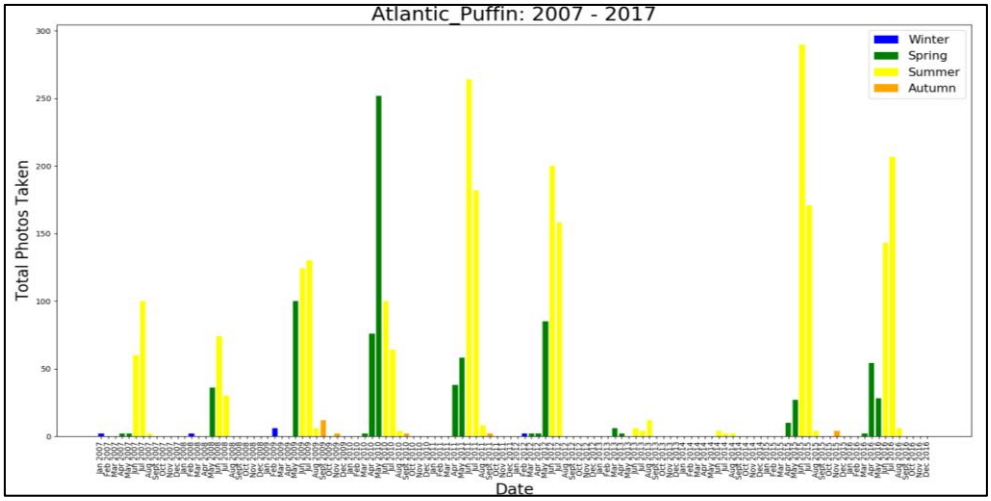


Figure 43 Results: Atlantic Puffin Seasonal Bar Chart 2007 - 2017

The Flickr data from 2007-2017 in Figure 43, demonstrates that the number of Atlantic Puffin photos are high in the early spring and summer months, and almost non-existent in the autumn and the winter months. This agrees with the initial research, that Atlantic Puffins migrate to the UK for breeding and raising fledglings. The reasons for the poor data counts in 2013 and 2014 are unknown, however it could be evidence of an event called the 'Atlantic Puffin Wreck' that occurred in 2013 and caused a notable dip in the puffin population. [49]

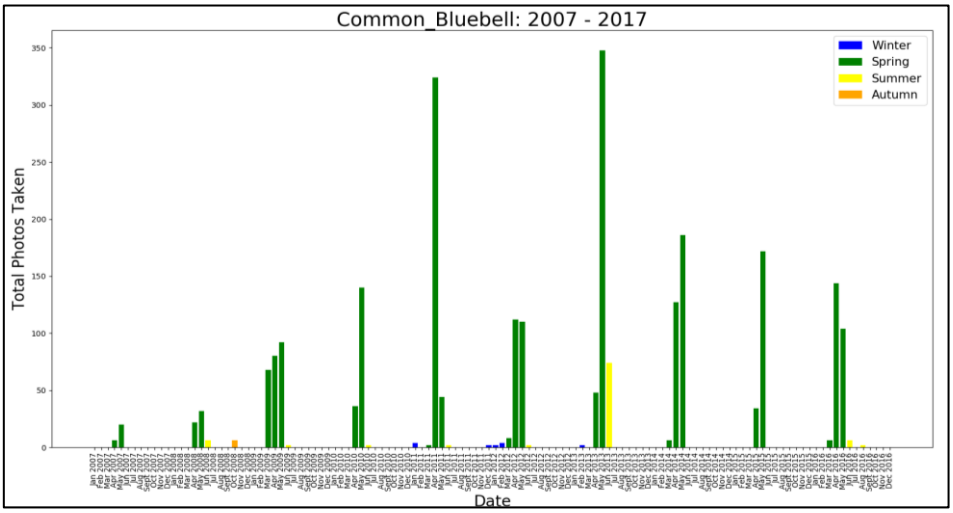


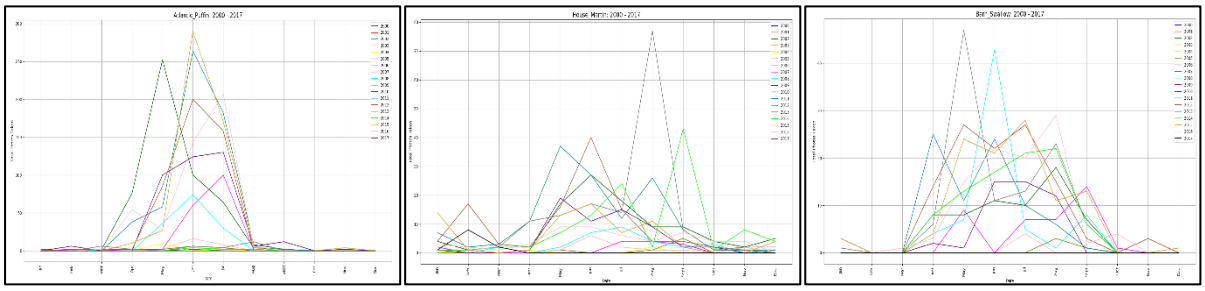
Figure 44 Results: Common Bluebell Seasonal Bar Chart 2007 - 2017

The graph above shows the monthly counts for the Common Bluebell, a plant that blooms in early spring and is gone by the summer. You can see that the Flickr data accurately displays this research as each of the significant plots are green which equals spring on the legend.

7.1.3 Yearly Line Graphs

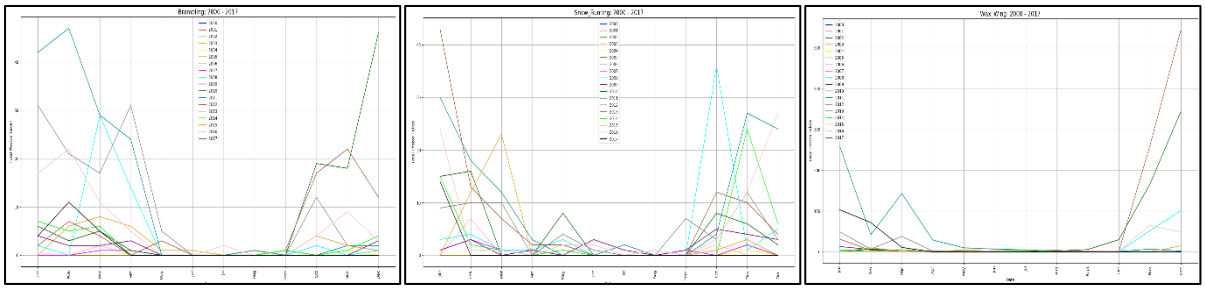
The year line graphs were made in conjunction with extracting data for the ten bird species. They provide a clear view of the points in the year that have high counts, making it simple to view migration patterns and further provide evidence of Flickr's use as a wildlife research tool.

Summer Residents (Atlantic Puffin, House Martin, Barn Swallow)



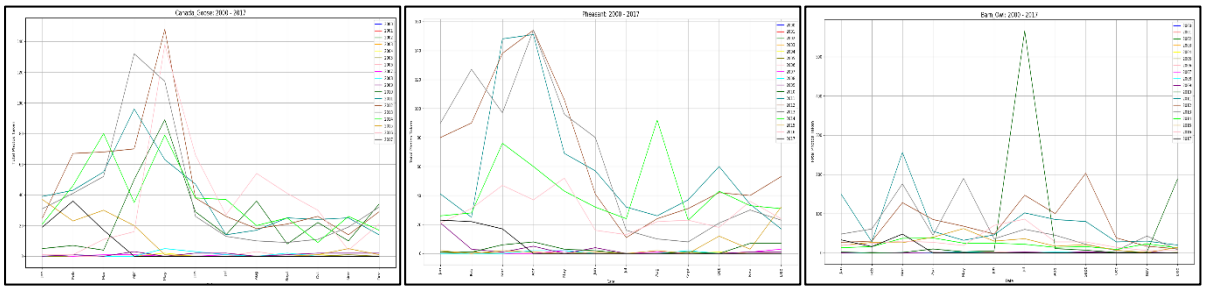
There is a clear trend in all three graphs stating that there are high numbers of these birds in the summer months, and little to none throughout the rest of the year. These birds' migration patterns have been accurately displayed using Flickr data.

Winter Residents (Brambling, Snow Bunting, Wax Wing)



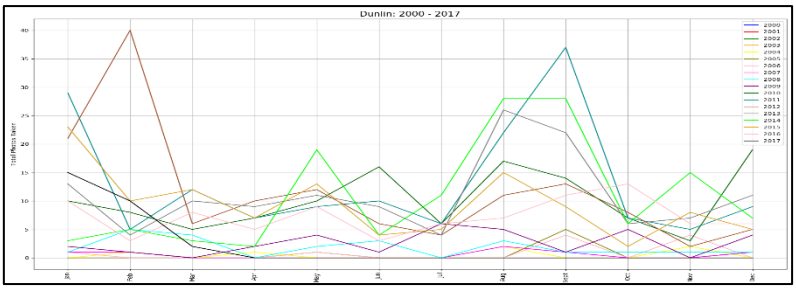
For the winter resident birds, there is again a clear trend. The trend shows high numbers of photos being taken in the winter months, and very little taken during the rest of the year. The Wax Wing in particular clearly shows a very distinct lack of photos during non-winter months.

Year-Round Residents (Canada Goose, Pheasant, Barn Owl)



For the resident birds, there is a trend suggesting that the numbers are level all year long. However, both the Pheasant and Canada Goose have recurring high counts in the first half of the year. The reason for this is most likely due to it being birthing season. During birthing season both types of bird will spend long periods on the ground with their eggs, this will make them easier to photograph and the prospect of eggs and hatchlings will attract amateur photographers. This trend not being visible with other bird species, could be due to them laying eggs in trees and hedgerows in nests out of view of the public.

Special Case Bird (Dunlin)



Initial research stated that the Dunlin can also be found all year round within the UK. This trend is visible above. As mentioned in the method section, the reason for selecting the Dunlin as a special case, is due to its recorded nature to migrate to different sections of the UK throughout the year, this type of behaviour will be better viewed on a map projection.

7.1.4 Timeline Evaluation

Flickr Geotagged Data Timelines:

These graphs were created to determine if a certain time of the year produced more geotagged photographs. The results of the total geotagged images timelines were very useful in identifying trends for the past, present and future use of Flickr as a research tool and certainly fulfilled the purpose they were created for.

The unnormalised total bar chart clearly shows a large spike in Flickr usage which peaked in 2012 but has declined steadily since. Formatting of the graphs is clear, as is the displayed data, in summary the graph establishes assurance for anyone considering the use of Flickr for their own projects in the future.

The two normalised graphs also very clearly establish a trend throughout the decade's use of Flickr, indicating that there are summer spikes in photographs and lows in the winter. The graphs are display clear data. To improve the graphs to further aid this specific project it would be useful to expand the use of tags to more than just 'wildlife', it would provide a much broader scope of the captured geotags if multiple specific tags were used such as 'bird', 'mammal', 'amphibian' etcetera.

Seasonal Bar Charts:

The season timelines were established to clearly demonstrate the seasonal variation of species. Based on the results, they effectively highlighted the season in which each species is most commonly found, such as Common Bluebell in the spring. The graphs are very clear in showing seasonal variation due to the colour coded bars and accompanying legend.

Yearly Line Graphs:

The yearly line graphs were created to provide evidence that the migrations patterns of 10 separate bird species could each be identified using Flickr data. The results confirm that this was the case. The graphs accurately displayed each bird's migrations and therefore provide evidence that Flickr can be utilised as a tool to track species behaviours.

7.2 Grid Maps

7.2.1 Visual Comparison

The visual comparison maps were created to enable comparison between the normalised Flickr and NBN data to detect trends that are impossible or very difficult to find by reviewing two textual lists of plots. Below details on several of the results are found. Note for the following figures the right map is NBN and left is Flickr.

Canada Goose:

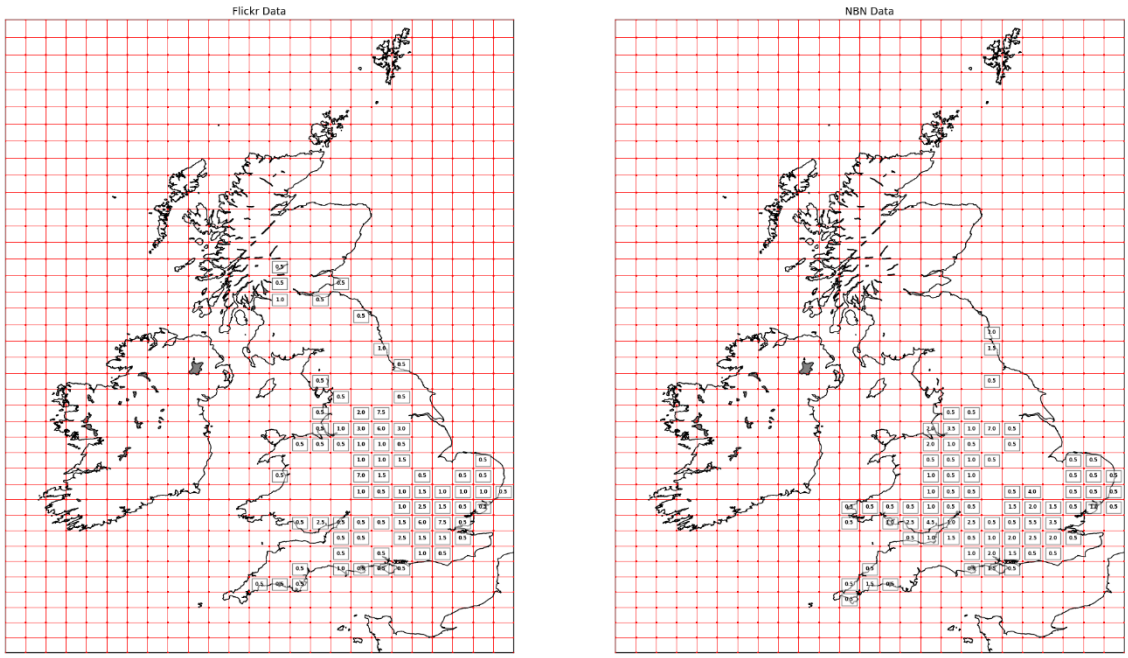


Figure 45 Results: Canada Goose Normalised Data Flickr (right) NBN (left)

The graph above shows the distribution of the Canada Goose using Flickr and NBN data. On first glance it is clear that the two projections are very similar, they both highlight that the highest populations can be found in Southern and Mid England (especially around Norfolk and London areas), with smaller populations in Wales and trace populations in Scotland. If the maps are studied in closer detail, it is evident that the highest populated cells for each map are in the same locations. For example, Flickr and NBN have 6% and 5.5% accordingly for the cell covering central London, and 7% and 6% for a cell just south of Leeds. This data would suggest that Canada Goose are most common in areas populated with people, such as city parks like Hyde Park, as initial RSPB research suggested.

Flick data also displayed its own unique trend for the Canada Goose. This trend showed that the highest populated cities in the UK (London, Birmingham, and Leeds) each had 7.5% population values. This would suggest that the highest populated areas are generating the most captured photos on social media. This may be due to there being more people taking photos. In this case, it does not damage the data's credibility for displaying known wildlife distributions as the NBN data agrees with Leeds and London's high species populations, and initial RSPB research confirms the species are attracted to cities.

Pheasant:

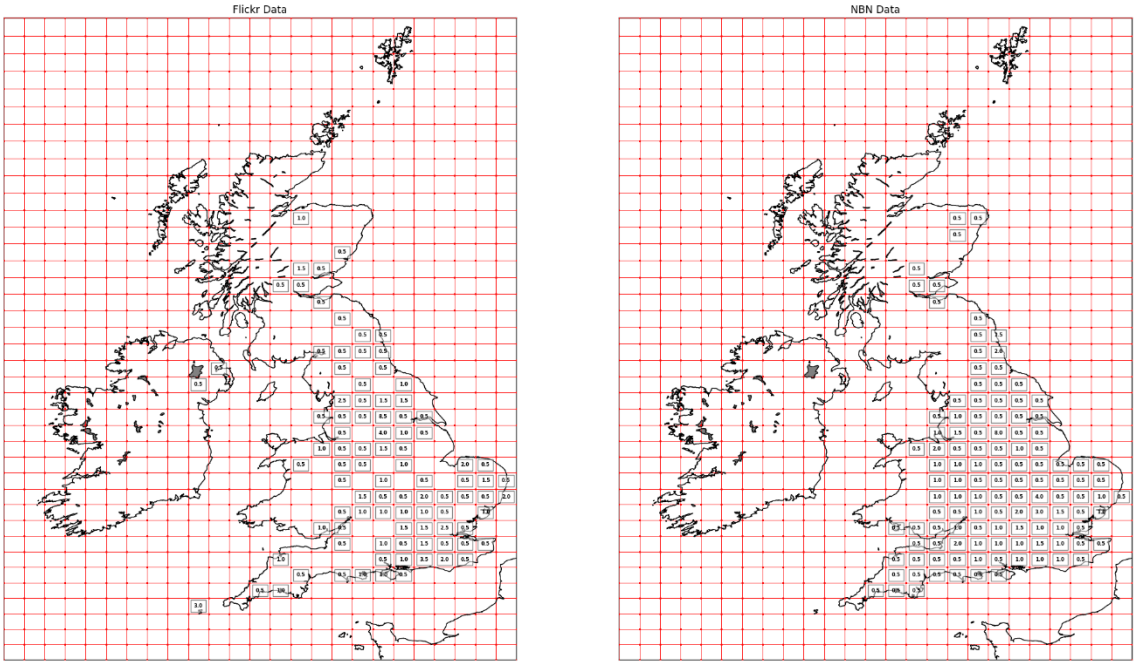


Figure 46 Results: Pheasant Normalised Data Flickr (right) NBN (left)

The graph above shows the distribution of the Pheasant using Flickr and NBN data. Once again visually the distributions displayed by the Flickr data correlates favourably with the ground truth data from NBN. Most Southern and Mid England is evenly dispersed, with few stand out populations, and excluding South Wales, there are few sightings in Wales and most of Scotland. Notably, the Flickr data is not as complete, however key features are still visible. For example, all regions of England contain populations including the most southern tip of West England, and Scotland and Wales have small populations. The Pheasant NBN dataset is the largest those studied in this project, therefore it is acceptable to expect the graphed data to appear so complete (with no gaps that are visible within the Flickr graph).

Notably the trend encountered with the Canada Goose Flickr data, that densely human populated cities contains high photo counts, is not visible within the Pheasant figure. This is useful as it provides evidence that the city Canada Goose populations were due to wildlife behaviours and not just human behaviour.

Atlantic Puffin:

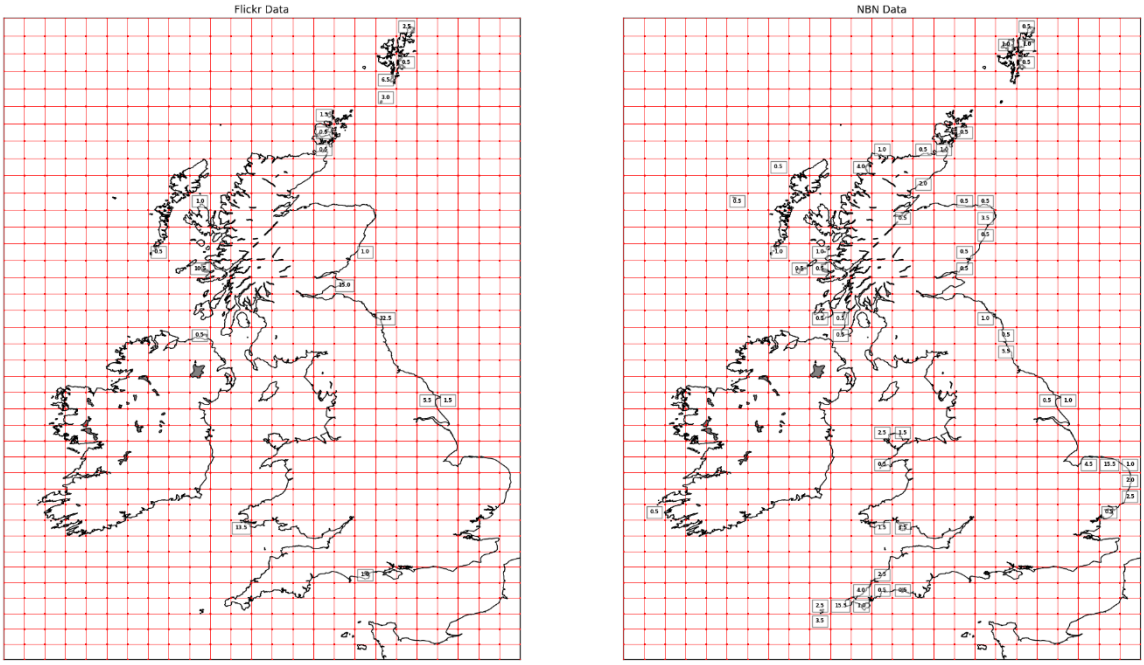


Figure 47 Results: Atlantic Puffin Normalised Data Flickr (right) NBN (left)

The graph above shows the distribution of the Atlantic Puffin using Flickr and NBN data. This is an example of Flickr data poorly displaying the distribution of wildlife in comparison with the ground truth data. The Flickr dataset shows only 18 separate cells with Atlantic Puffin sightings, whereas the NBN data appears in 51 cells throughout the UK. Flickr shows no record of Atlantic Puffins in large areas of the UK where NBN states there are many, such as the Cornwall and Norfolk.

The Flickr dataset also displays cells with very high percentages of the UKs puffin sightings. For example, known Puffin breeding sites like Skomer Island (Wales), Isle of May and Lunga (Scotalnd), and the Farne Islands (England) have 13.5%, 15%, 10.5% and 32.5% respectively of the total captured photographs. This is particularly interesting as it suggests amateur photographers looking to take photos of Atlantic Puffins will congregate at known breeding sites, rather than risk a less likely encounter on the coastline. Although this data does not accurately show Atlantic Puffin distributions, it does highlight a social/cultural trend of the public to visit certain sites.

Species Name	Flickr Cells	NBN Cells
Atlantic Puffin	***18***	***51***
Barn Owl	71	102
Barn Swallow	69	103
Brambling	74	69
Canada Goose	75	79
Dunlin	78	64
House Martin	80	87
Pheasant	93	117
Snow Bunting	45	34
Wax Wing	***79***	***44***

Figure 48 Results: Total Populated Cells per Species

Wax Wing:

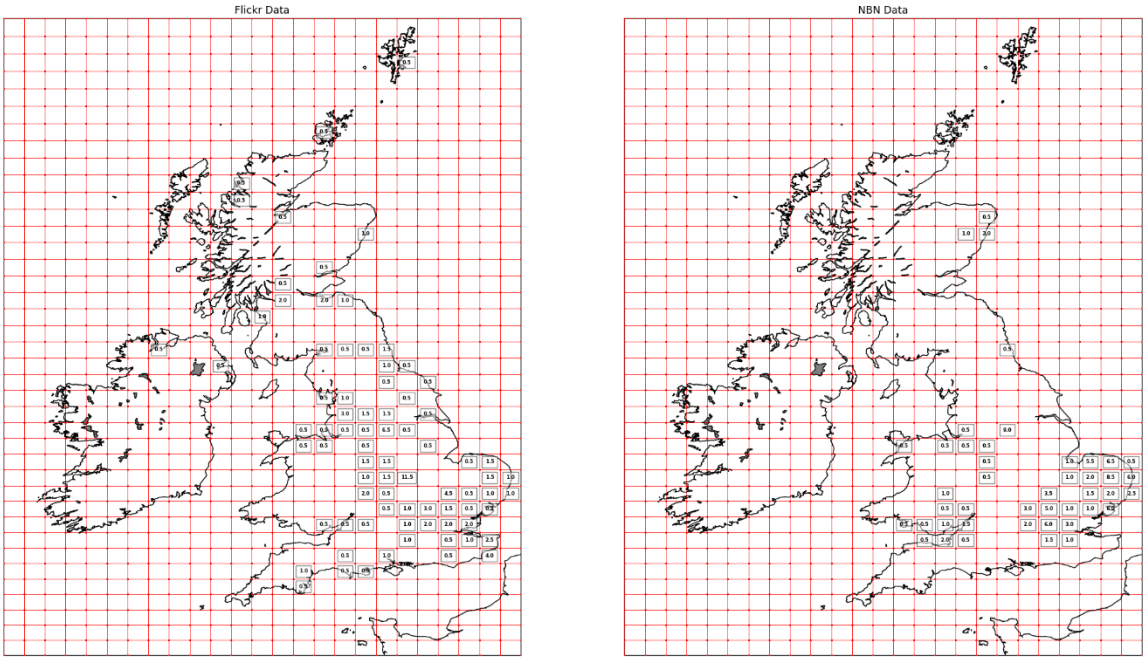


Figure 49 Results: Wax Wing Normalised Data Flickr (right) NBN (left)

The graph above displays the distribution of the Wax Wing using Flickr and NBN data. The Wax Wing's datasets are particularly interesting, this is due to the Flickr data set having considerably more populated cells than the NBN data (79, 44). Due to the large quantities of NBN data it is common for the population to be more dispersed than the Flickr set. The difference between the two datasets suggests that, the Flickr dataset is too small and therefore the normalisation is removing fewer small sightings. This is the opposite issue to the one displayed by the Atlantic Puffin data, in which there are many densely-populated areas causing smaller sightings to be normalised off the map.

The most densely populated county in the ground truth data is Norfolk/Suffolk, which can be seen in the Flickr data also. This trend makes sense as Norfolk is the most eastern point of the UK, and the Wax Wing migrates to the UK from Russia and Scandinavia in the East. However, the rest of the England shows many Flickr sightings that are false when compared with NBN.

7.2.2 Quantitative Evaluation of Results

Species Name	Recall	Precision	F1 Score	Accuracy
Atlantic Puffin	0.14	0.388889	0.205882	0.851648
Barn Owl	0.392157	0.56338	0.462428	0.744505
Barn Swallow	0.38835	0.57971	0.465116	0.747253
Brambling	0.376812	0.351351	0.363636	0.75
Canada Goose	0.632911	0.666667	0.649351	0.851648
Dunlin	0.5625	0.461538	0.507042	0.807692
House Martin	0.45977	0.5	0.479042	0.760989
Pheasant	0.675214	0.849462	0.752381	0.857143
Snow Bunting	0.470588	0.355556	0.405063	0.870879
Wax Wing	0.590909	0.329114	0.422764	0.804945
AVERAGE	**0.468921**	**0.504567**	**0.471271**	**0.80467**

Figure 50 Results: Similarity Calculations

Recall Results: Figure 50 indicates that the average result for the recall was 0.4689. Demonstrating that on average, 46.9% of all ground truth NBN data was also reflected by the Flickr data. Over half of the ground truth data is not displayed or reflected by the Flickr data, meaning that on average Flickr data is missing over half of the wildlife locations that are necessary to get a full picture of the species behaviour.

Comparing specific species the Atlantic Puffins recall is considerably worse than the rest, which was evident from a visual comparison and is caused by amateur photographers visiting specific sites for their photos of Puffins. Only 14% of all the Atlantic Puffin distribution in the UK was picked up by Flickr, therefore puffins clearly cannot be accurately researched using Flickr.

On the other end of the spectrum, are the Pheasant and the Canada Goose, two species that provided good results when compared visually with the ground truth data. Here, 67% of the Pheasants locations within the UK were correctly identified using Flickr data, as were 63% of the Canada Goose locations. This is considerably better than the Atlantic Puffin, and would certainly contribute to understanding the species behaviour.

Precision Results: The results of the precision calculations show that, on average, of the 46.9% of events that were correctly recalled by Flickr, 50.4% of the Flickr datums true cells were correct observations (relative to the NBN data), the other 49.6% gave false positive results. Therefore 49.6% of the 'true' Flickr results were in fact false in that they were not matched in the ground truth dataset.

Observe the Wax Wing the precision results, these confirm what the visual comparison indicated previously. Due to the Flickr data set containing significantly more results than the NBN equivalent, 59% of the ground truth is recalled, which is above average. Only 32% of the total true Wax Wing

data set is included, this means that every time Flickr states a cell contains a Wax Wing, 68% of the time it is incorrect.

The top result, in terms of precision, is the Pheasant. The pheasant correctly recalled 67% of the Flickr data using 84% of its data. Demonstrating that only 16% of the pheasant true data is not also reflected by the NBN data. The Canada Goose drops slightly, as only 66% of its data reflects true distribution, 34% of its true data is false. This is a good example as it demonstrates that just because the recall of a dataset is good, it doesn't mean the precision is (in comparison with the Pheasant results).

F1 Score Results: The F1 Score is the harmonic mean of the precision and recall. Due to the difference between recall and precision in some cases, such as the Wax Wing, it is important to find a mean between to two to have a more accurate classifier. The average F1 Score is 47.1%, the harmonic mean of 46.9% (recall) and 50.4% (precision). On average the F1 classifier isn't too different to the original classifiers. However, in the case of the Wax Wing and Atlantic Puffin the new classifier is significantly different and provides a fairer representation of the data set using a single figure, as opposed to using just precision or recall.

Accuracy Results: Accuracy is used to find the percentage of cells that matched (those that agreed that there was or wasn't data). The results in the table show that the average accuracy of the Flickr datasets is 80%, which is a good result. However, the reason for the accuracy being this high is due to the number of true negatives with each species. Flickr and NBN agree that there are zero species sightings in many places of the UK, which is understandable, as most species only appear in certain habitats. As displayed by the data, it is possible to have a high accuracy, but low recall and precision.

Atlantic Puffin		Flickr	
		True	*False*
NBN	*True*	7	43
	False	11	303

Barn Owl		Flickr	
		True	*False*
NBN	*True*	40	62
	False	31	231

Barn Swallow		Flickr	
		True	*False*
NBN	*True*	42	64
	False	27	231

Brambling		Flickr	
		True	*False*
NBN	*True*	26	43
	False	48	247

Canada Goose		Flickr	
		True	*False*
NBN	*True*	50	29
	False	25	260

Dunlin		Flickr	
		True	*False*
NBN	*True*	34	27
	False	44	259

House Martin		Flickr	
		True	*False*
NBN	*True*	41	48
	False	39	236

Pheasant		Flickr	
		True	*False*
NBN	*True*	79	38
	False	14	233

Snow Bunting		Flickr	
		True	*False*
NBN	*True*	16	18
	False	29	301

Wax Wing		Flickr	
		True	*False*
NBN	*True*	26	18
	False	53	267

Figure 51 Results: Confusion Matrices

R-Squared Results:

Species Name	*Atlantic Puffin*	*Barn Owl*	*Barn Swallow*	*Brambling*	*Canada Goose*	*Dunlin*	*House Martin*	*Pheasant*	*Snow Bunting*	*Wax Wing*	***AVERAGE***
R-Squared	0.000185	0.01161	0.026599	0.125543	0.38706	0.23738	0.047376	0.2452	0.205181	0.124618	**0.141075**

R Squared provides another method for determining the similarity between the Flickr and NBN datasets. You can see that the average score is 0.141 or 14.1%, although this may inherently seem like a low value the R Squared results are not perfectly determined by the magnitude of the number. For example, when studying human behaviour, it is typical for results to be less than 50% due to humans being hard to predict, this value is relevant as the main source of Flickr data depends on the willingness of the general public to take photos against variables such as weather and opportunity, and furthermore the species to be available and visible for capturing.

Despite these reasons few of the datasets still provide poor results by any measure. Once again, the poor quality of the Atlantic Puffin data for displaying the bird's distribution is apparent. An interesting result to note is that again the Canada Goose and Pheasant are significantly above the average result once again, further proving their accuracy.

Hellinger Results:

Species Name	*Atlantic Puffin*	*Barn Owl*	*Barn Swallow*	*Brambling*	*Canada Goose*	*Dunlin*	*House Martin*	*Pheasant*	*Snow Bunting*	*Wax Wing*	***AVERAGE***
Hellinger	0.94097	0.74683	0.70208	0.74683	0.526905	0.640539	0.68563	0.488321	0.697971	0.70148	**0.687758**

Finally, Hellinger is also used to evaluate the similarity between two datasets. These calculations have been included to provide another view of the Flickr datasets in comparison with NBN, and to assess if it produces similar results to the confusion matrix and R-squared results. Note that Hellinger results are between 0 and 1, and that a lower value the more similar the datasets.

Review the table below and see that Hellinger, R Squared and F1 Score rank 20% of the species the same, and a further 30% agree within 1 rank unit. This is useful as it reinforces the results, as three unique comparison calculations agree with 50% of results being within 1 rank unit of each other. However, it can be noted that F1 Score and Hellinger have greater similarity than each of them compared with R Squared.

Species Name	F1 Score	Hellinger	R Squared	Average
Atlantic Puffin	10	10	10	*10*
Barn Owl	6	8	9	*9*
Barn Swallow	5	7	8	*7*
Brambling	9	8	5	*8*
Canada Goose	2	2	1	*2*
Dunlin	3	3	3	*3*
House Martin	4	4	7	*4*
Pheasant	1	1	2	*1*
Snow Bunting	8	5	4	*5*
Wax Wing	7	6	6	*6*

Figure 52 Results: Similarity Calculation Rankings

7.2.3 3D Maps

The 3D maps are used to view the count values on a grid map, rather than using a number. The method allows for an interactive view of the data, providing opportunities to locate and focus on certain areas from many angles. While the technology is interesting, the derivable results are not any more useful than a 2D map and in some cases are significantly worse.

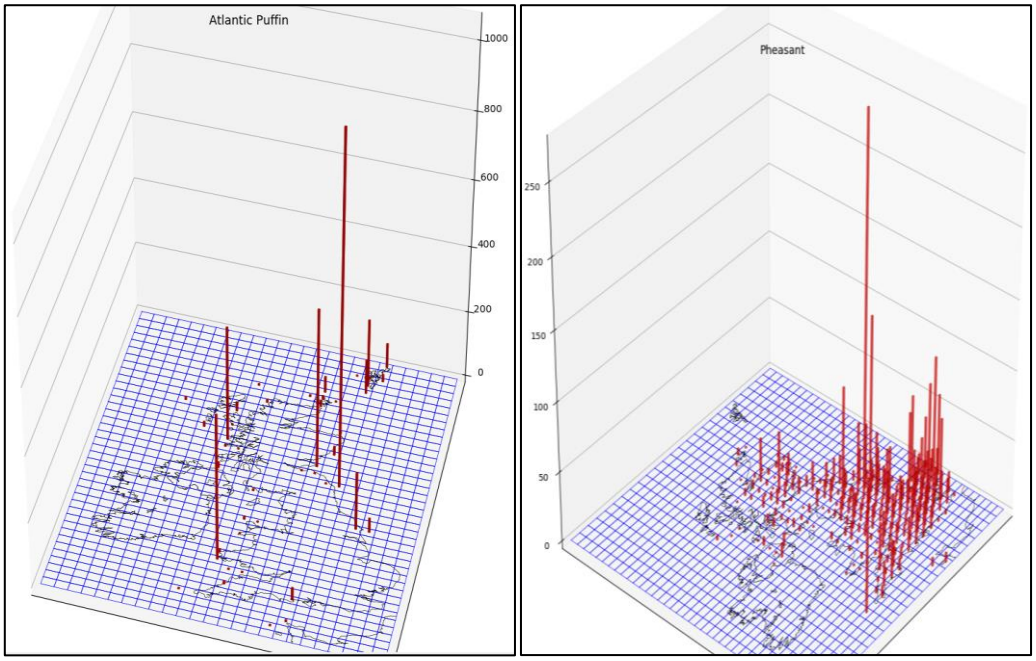


Figure 53 Results: Atlanitc Puffin 3D Map (left), Pheasant 3D Map (right)

Above are two figures showing the datsets for the Atlantic Puffin and Pheasant respectively. The Puffin example provides some use as it is clear to see each column and what cell it is connected to, therefore it is visible that the popular breeding sites have a large amount of data (although it is not possible to see exactly how many photos have been taken). The pheasant example is inadequate, due to the map being so populated it is not easy to see which cell each column belongs to, and large columns block other columns from view (despite the transparency of the columns).The only trend that can be seen, is the higher quanity of photo captured in the South compared with the North.

7.2.4 Grid Map Evaluation

Visual Comparison:

The visual comparison was created to evaluate the similarities between the Flickr and NBN data, with reasoning that cannot be replicated by computational or mathematical methods. The visual comparison was useful as it allowed a rationale for empty cells in the Flickr data to be provided, and took the knowledge of there being significantly less Flickr data available into account. Overall, the visual comparison was useful not only to assess the similarity, but also to identify and further research unexpected results. For example, the Atlantic Puffin hotspots.

Quantitative Evaluation:

The similarity calculations were created to compare the datasets using various standard methods. The comparisons were useful in uncovering an ‘official’ value to confirm the similarity, and the

ranking being 50% similar across the three comparison methods, validates the results are trustworthy. To improve this, section the accuracy values could be excluded as they don't provide an accurate measure of the similarity. Also, a fourth comparison method (such as KL divergence, or Earth Movers distance) could be included to assess if the slightly dissimilar R Squared results are an outlier, or if other methods produce results that more closely resemble them.

3D Mapping:

3D mapping was not useful in assessing either data set as it didn't provide any further use than the 2D data. In the future 3D mapping will not be used.

Grid:

The grid was used to help identify the total counts in specific areas of the UK. Before the gird was implemented data was displayed using normal plots, however this gave no indication of just how many photographs have been taken in certain areas. For example, Skomer Island has over 100 photographs captured of Atlantic Puffins, but using scatter plots, it appeared as 1 or 2 overlapping plots due to the small scale of the island and the close vicinity of the captured photos. The grid was also vital in completing all the similarity calculations, without which it would not be possible to compare the data using any useful and known methods.

The grid could be updated to improve its accuracy. Currently the grid is projected using latitude and longitude values, however due to the curve of the earth, longitude and latitude values equal angles apart do not create a square shaped cell. Therefore, to create a square shaped cell, the distance between rows was manipulated to form a square shaped cell. To improve this method, a grid reference system, such as UTM, would be used to plot the grid instead.

The grid could also be more efficiently used to retrieve the cells data if a different geo spatial database was used. Currently to find data in each cell an exhaustive search of the database is performed to obtain the total photographs. This is very inefficient and prevents the use of a grid with hundreds of columns, as it would take hours to create one graph. This was not an issue as the datasets were too small to require so many cells. To improve on this method, a geo spatial database could be used to better store and query geospatial data using cell IDs.

7.3 Map Projections

7.3.1 Time Slices

Time slice maps were created to see if Flickr data could uncover species movement across the UK. The focus for these figures was bird and plant life to confirm known migrations and blooming research. However, once it was apparent that the graphs could be used to view trends, research on mammals, reptiles, and amphibians was continued with hopes to see interesting trends. The graphs show all the data for each species from 2000 to 2017 for each month.

House Martin:

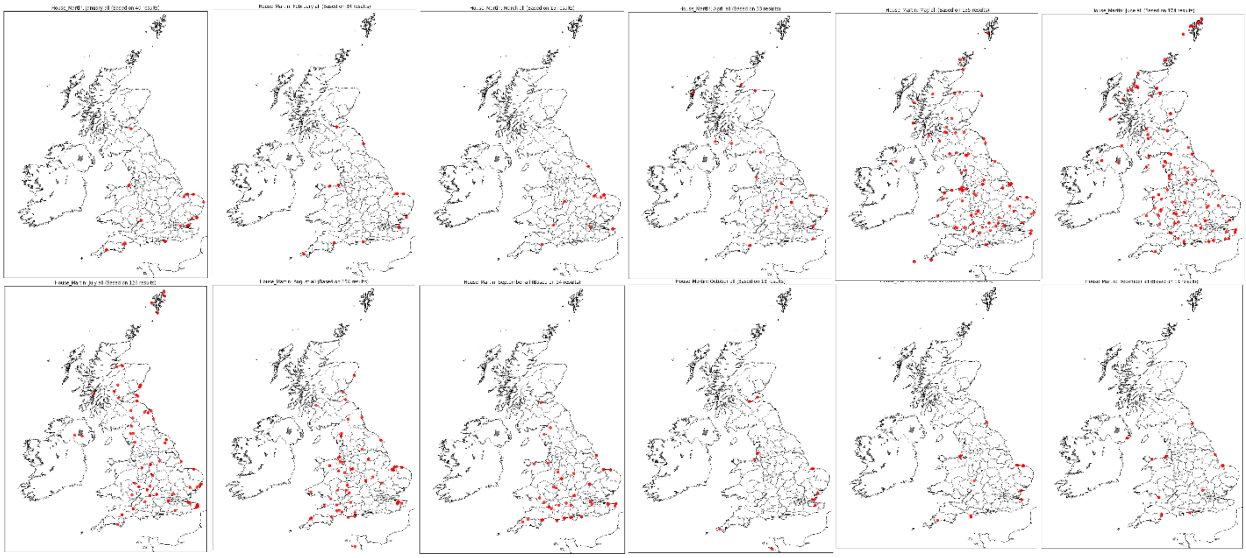


Figure 54 Results: House Martin Time Slices

The House Martin was the best performing summer resident bird in the similarity calculations, therefore it is best suited as an example to accurately show the South to North migration pattern. The House Martin spends the summers in the UK, however it spends the winter in Africa. This migration is clearly displayed above, it can be seen that the number of plots sharply increase from April to May (309% increase) as the birds start to arrive, the birds numbers sharply decrease from September to October (80% decrease). It is also possible to visually recognise a trend from March to October of the birds arriving in the South of the UK around March (closest to Africa), moving up the country around May, moving back down again around August, and leaving in October. The winter months then provide very few sightings.

Snow Bunting:

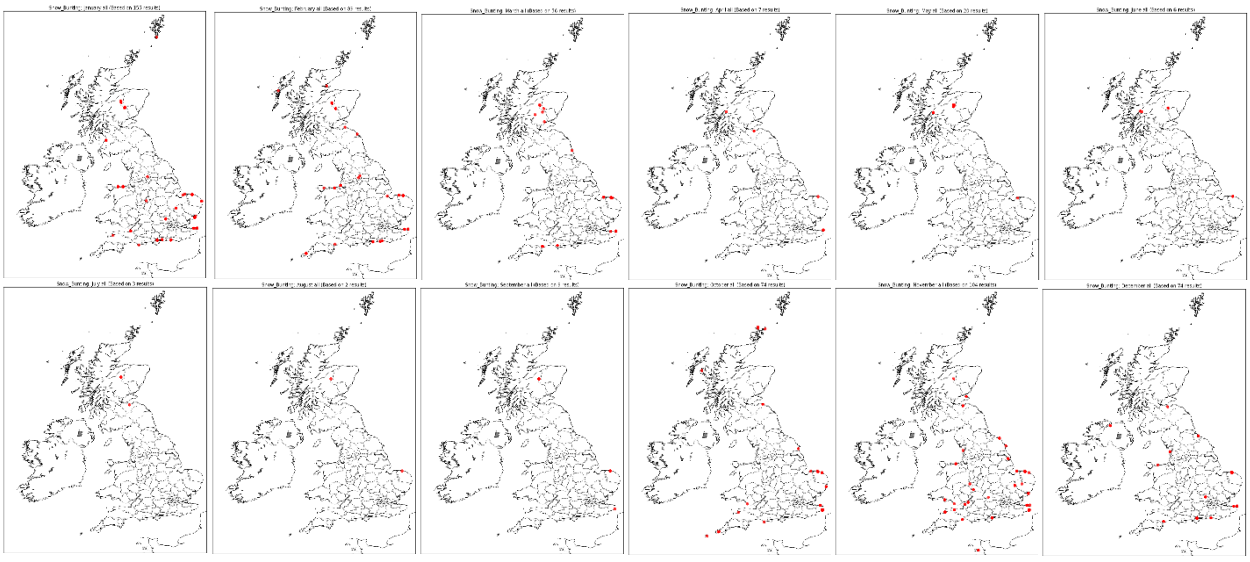


Figure 55 Results: Snow Bunting Time Slices

The Snow Bunting performed poorly in the results of the similarity calculations with NBN, however, when displayed in time slices it clearly shows an East to West migration pattern. The Snow Bunting spends the winter in the UK, however during the summer it is found in Scandinavia. This migration pattern is evident above as there is a sharp increase from September to October (722% increase) when it gets colder and the birds arrive. There is then a sharp decrease from March to April (87.5% decrease) as it starts to get warmer. By visually analysing the maps it can also be seen that the birds arrive on the East coast (closest to Scandinavia) in October, then spread out across the UK by January, returning to the East coast by March on their migration back to Scandinavia. As research from RSPB suggests, there is also a small population of Snow Buntings in Scotland through the summer. Whilst the Snow Bunting data doesn't compare well with the ground truth, it does fit the migration expectations based on RSPB research.

Grass Snake:

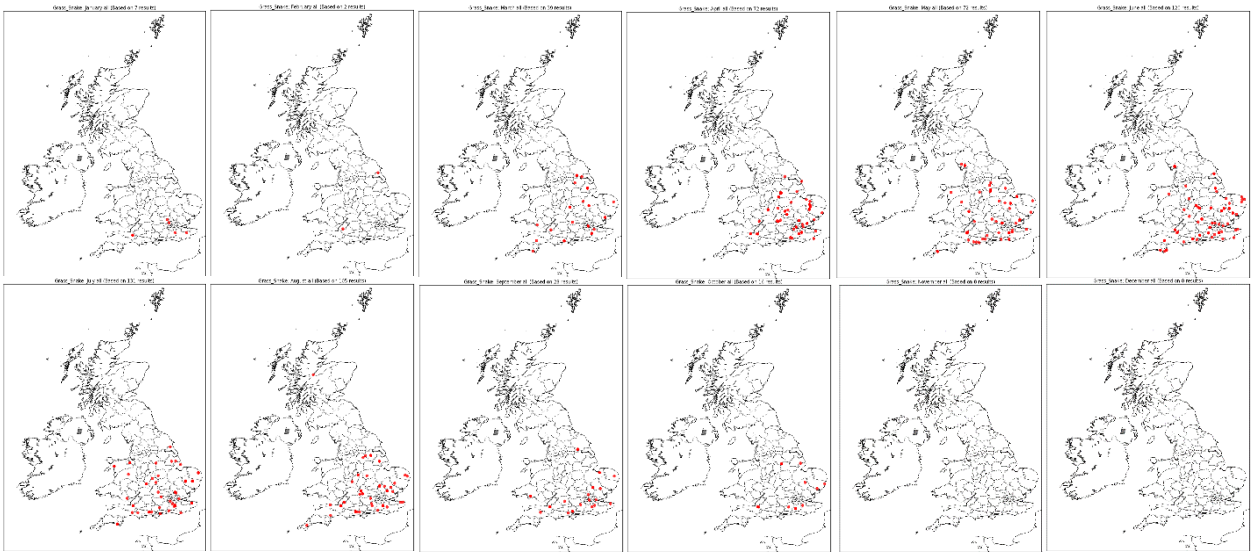


Figure 56 Results: Grass Snake Time Slices

Grass snakes, like all reptiles, hibernate during the winter (more specifically from October to as late as April). This hibernation pattern can be seen above as the numbers start to drop in September (73% decrease) then by November/December not a single photo has been uploaded in the time range. The number of photographs then starts to increase March/ April, and continuing to an all-time high of 120 in June. By visually analysing the time series the hibernation can be visualised as the winter months have little to no plots.

Based on the position of the plots it can be concluded that the Grass Snake favours the south, this is most likely due to the temperature being hotter and better suited for a reptile.

Common Frog:

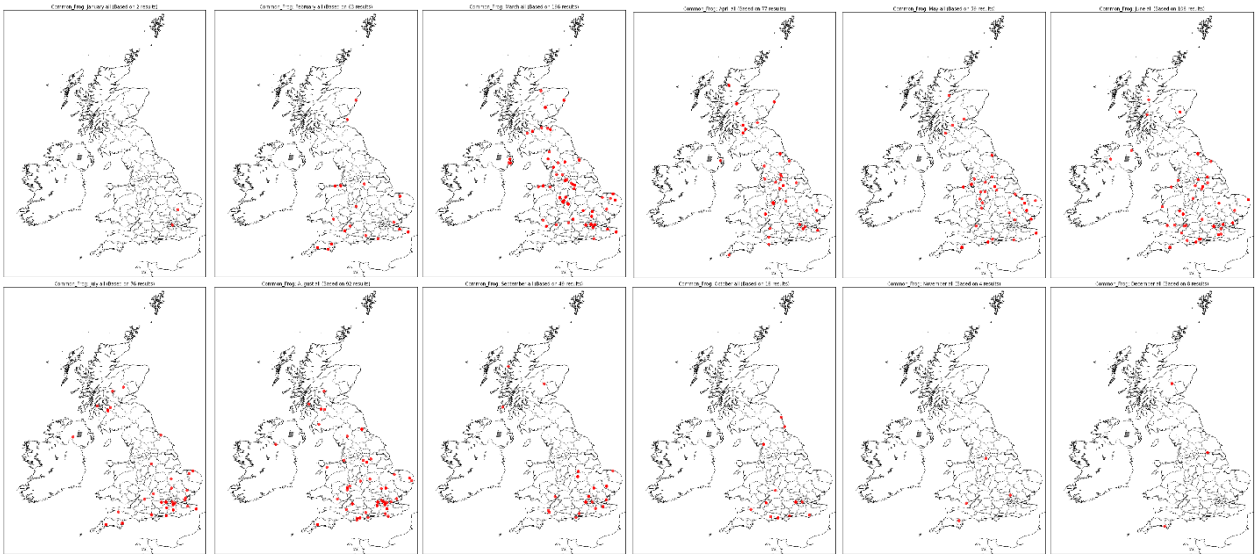


Figure 57 Results: Common Frog Time Slices

The Common Frog is also a species that hibernates during the winter. Like the Grass Snake, this hibernation can clearly be seen in the time series above. Visually it can be determined that the sightings decline as summer ends and winter begins, with only 4 and 8 documented sightings of common frogs in November and December respectively since 2000. The Common Frogs behaviour is clearly well observed using Flickr, not only to demonstrate its hibernation pattern but also providing a very good estimation of the species range.

Common Bluebell:

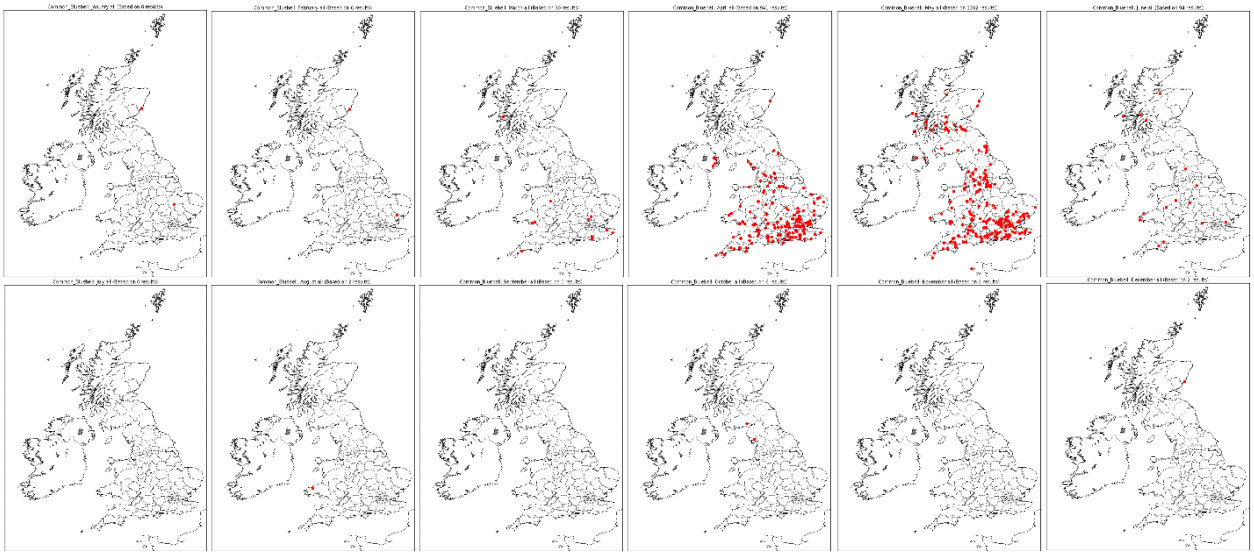


Figure 58 Results: Common Bluebell Time Slices

The Common Bluebell is a species of plant that flowers March to May. You can clearly see the flowering range of the Common Bluebell, there are little to no sightings of the flower in the summer, autumn, and winter, however in spring there are thousands of sightings. From March to April there is a 945% increase, which suggests that the plants are at their most common from April to May (941, 1262). Based on this data it can be determined that Flickr is a useful tool for researching flowers such as the Common Bluebell.

7.3.2 Dunlin Data

The Dunlin has been described in this project as a 'special case', this is due to it being a species that migrates to different sections of the UK throughout a year for breeding and hatching purposes. It was very interesting to determine if Flickr data could display such specialised behaviour. The Dunlin is rank 3 in quantitative evaluation experiments which shows the data reflects the ground truth data above average levels.

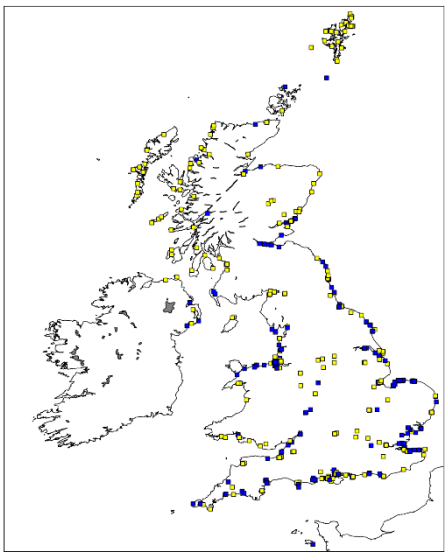


Figure 59 Results: Dunlin All Data

The bird spends winters along the coast around England, Wales, and East Scotland, and it spends the summer in mountain areas of Central England and East Scotland and Shetland.

The graph to the right displays summer sightings in yellow and winter sighting in blue. Demonstrated is a trend of winter sightings around the coast, and summer sightings in the mountain areas and Shetland. Visual comparison of Flickr results with the RSPB map is very good, the internal UK migration is clearly visible as researched. The blue does not cover as much coast as the RSPB map, this will be due to much of the coast not being easily accessible to the public.

7.3.3 Map Projections Evaluation

Time Slices Evaluation:

The time slice graphs were implemented to physically show where the species were moving to and from over the course of a year. While the timelines show high levels of species, they do not indicate where the species are. These maps were created with hopes to view species movement across the UK, by plotting data for summer and winter birds can be seen moving in and out the country in monthly time slices. It was also possible to view the flowering of wild plants, and hibernations of reptiles and amphibians. The time slices graphs clearly highlighted that Flickr can be used to view migration, hibernation, and flowering, even with a small dataset.

Dunlin Data:

The dunlin results provide excellent proof that Flickr data can be used to track species movement within the UK. The graph shows that the Dunlin spend the winters around the coast, and the summers in the highlands with just 991 data points, confirming initial research. This results suggests that future trends highlighted using other species data are worth being research specifically.

8.0 Conclusion

The aim of this project was to use geotagged images from social media to analyse the behaviours of UK wildlife, to determine if Flickr could confirm the existing knowledge of UK migration and hibernation, and be used to identify new behaviour. Data was extracted from the social media platform Flickr as it seemed the most appropriate for the requirements of pinpointing the exact location the image was taken. Platforms like Twitter are usually opinion based meaning that often people will tweet about an animal from their homes to perhaps give an opinion on an animal's welfare, or the cuteness of a species. In brief, the tags associated with these opinions would cause too much distortion in the data, whereas Flickr geotags are more likely to highlight where the photo was taken.

During research, it was decided to explore species of different classes to produce a diverse range of results. A diverse range would present a more complete conclusion of how well social media can be used to study wildlife. To research the data for these species they were plotted on a UK map in bulk and in time slices. The results of this research confirmed that Flickr could be used to show hibernation in reptile and amphibian species, migrations of birds and flowering of plants as displayed by Common Bluebell. There is also the emergence of the Orange Tip Butterfly, which can be used to indicate the start of spring (See appendix, Figure 5). However, while the data appeared interesting and the aim to analyse existing wildlife knowledge seemed complete, there was no way to officially measure the accuracy of Flickr data.

After further research, it was also decided to research several additional bird species with different migration habits to determine if Flickr could accurately display several known behaviours. To evaluate the accuracy of the bird migration data it was compared visually with ground truth data from NBN, and also via standard similarity calculations that gave a formal measure of Flickr accuracy. The results concluded, that whilst certain species data provide an accurate depiction of the real-life behaviour, certain species data was distorted by issues that arise from the use of social media data.

There were examples of high populations affecting the number of results in an area. For example, the Canada Goose has very high photo counts in London, Leeds, and Birmingham which are the three most populated cities in the UK. While the ground truth did agree with London and Leeds, it had very few sighting in Birmingham, which suggests that the Canada Goose is a popular species to photograph in the city despite their numbers not being high (as reflected by NBN).

There was also clear evidence of the public congregating to known species breeding ground to take photos. The clearest example displayed in this project is the Atlantic Puffin as its full UK distribution was very poorly displayed. This is due to 71.5% of all Puffin sightings being recorded from just 4 different breeding sites.

Furthermore, there is evidence of people taking photos of certain species in higher numbers during different times of the year. For example, the monthly total photos of Canada Goose and Pheasant spike during their breeding season in April / May when the birds spend more time on the ground to protect and incubate their eggs, and when the hatchlings are visible.

Moreover, there is evidence of people congregating in known breeding grounds at certain times of the year. The Grey Seal results showed that 44.5% of all photos are taken at Donna Nook National Nature Reserve (See appendix, Figure 6), and peak time for Grey Seal photos is during November when the Grey Seal pups are born and visible on the beach. It can be concluded from these results, that the public are attracted to sites with higher chances of spotting the species, and further attracted during breeding and birthing seasons.

Research into the total number of UK geotagged images on Flickr also yielded interesting results of the publics photography behaviour. The results concluded, that the summer produces double the number of captured photos when compared with the winter, this suggest that factors such as the weather, school holidays, and workers using their annual leave affect the number of photos being taken. Further research of total geotagged photos in the UK with the tag Wildlife highlighted an even larger difference between winter and summer photo levels. These statistics will be affected by several more species being visible in the Summer (as reptiles and amphibians are not hibernating).

There are many factors which can distort the value of Flickr data for viewing known wildlife behaviour, however many species are still not distorted past being useful (such as Pheasant and Canada Goose). The data that is distorted by popular viewing spots, breeding seasons, and the weather are still useful for conservation as it clearly demonstrates the general public's viewing habits of species. This knowledge clearly has potential to determine how human interference affects animal behaviour, or to decide when best to provide the public with advice on viewing particular species and where best to display that information.

To conclude, the project was a success. Produced results certainly show how social media can be used to research animal behaviour, proof has been provided to suggest that Flickr can be used to track known behaviours, and be used to reveal how the public are drawn to specific areas at specific times. The results of the project are useful and relevant to computer scientists and wildlife enthusiasts alike, furthermore it is clear that there are many ways in which the project can be developed in the future.

9.0 Future Work

There are many ways in which this project could be further developed in the future. This project has put much work into exploring multiple ways in which the Flickr data can be used, however there is room for each individual aspect to be narrowed down and researched in greater detail.

Further work can be split into either a wildlife and conservation, or computer science driven approach . Wildlife and conservation driven would be more focused on analysing the trends and data already uncovered, by means of further research and determining reasons for outlier datum presence (for example a grass snake spotting in Northern Scotland, or Common Bluebell appearing early one year). Whereas computer science driven approaches would focus on using more advanced methods of comparison and expanding the dataset to allow for more accurate and complete analysis.

9.1 Computer Science Driven Future Work

The first section that can be expanded in the future would be the dataset. While Flickr has exclusively been used for this project, there are other platforms available to expand the size of the dataset. For example, Instagram and Twitter could both provide additional geotags to the dataset. By expanding the dataset, it would be possible to view distributions across the UK more accurately and if data was more concentrated around highly species populated areas there would be opportunity to implement further comparison methods.

Another section that can be expanded with future work is the scope. Currently the project has only looked at the UK, however social media data is available for the whole world. Therefore, as a logical next step the dataset could be expanded to Western Europe, or the whole continent. This would open opportunity to study a much larger range of species, and for species such as the Snow Bunting it would be possible to analyse the whole breadth of its migration from Scandinavia to the UK and back.

To expand the dataset or the scope, a large part of the future development would be to create a database better suited to large amounts of geo spatial data. Due to the small size of the datasets a simple MySQL database could be used and graphs could still be produced quickly, however on occasion the processing time was still over 10 minutes (especially when creating multiple graphs at once). Therefore, any future expansion would require a spatial database that would not need to be exhaustively searched to return results for single cells. Programs such as Quantum GIS could be used to process geo spatial data and automate map production and work with the spatial database more effectively than Python.

Another aspect of computer science driven future work would be to implement some advanced geo spatial methods. It was initially planned to use Kernel Density Estimation (KDE) to create clusters, then use time slices to visually compare and note any related clusters that could suggest some form of internal UK migration. Given more time, KDE could be implemented to provide a better visualisation of bird species moving across the UK during their migration, and attempt to visualise the Dunlins internal UK migration. To further expand on visually comparing clusters, it would be an option to implement automated cluster comparison between time slices to automatically track migration patterns.

9.2 Wildlife and Conservation Driven Future Work

In this project, the assumption that NBN data is the ground truth was made for purposes of determining the accuracy of the Flickr data, and all Flickr data that isn’t reflected by NBN is incorrect. However, each of the Flickr geotags represents a perceived sighting of the species and if it does not

correlate with NBN, it could be documented as a new sighting of the species rather than incorrect. A future project for a wildlife researcher could involve, documenting these 'incorrect' sightings and determining if the sighting is legitimate (or incorrectly identified) by viewing the image via the stored URL. Any legitimate sightings that are not compatible with current understanding could then be researched to further comprehend the behaviour of a species and possible factors that may cause it to behave uncharacteristically.

Future work could also be completed to aid Conservation and Education. The project results, have determined a number of common photographer behaviours, such as congregating to known species breeding grounds, or taking high number of photos during birthing season. To educate the public on conservation and ensure they do not disturb animals, especially young, it would be possible to use Flickr data to best decide where to display public information. For example, the results of the South Wales Cultural Case Study determined an exact spot in Rhossili Bay where many photos are taken. The results of this project determined a specific area in which 44.5% of seal photos are taken, proving it is possible to pinpoint an exact spot to provide educational documents, or have a species expert present to ensure the species are treated with respect. Well placed knowledge could also be used as a platform to provide advice and knowledge, and spark public interest in the species and its conservation.

10.0 Reflection on Learning

This project has helped to utilise the skills and apply the knowledge that I have learnt during my time at Cardiff University. It allowed me to apply methods that were only theorised in lectures, and provided opportunity for me to utilise the development skills I established through hard work in Cardiff and on my placement. Also, it stood as a final challenge to apply everything I have learnt in one large project without the support of a group and following my own originality and design. Also, it provided opportunity for me to combine two fields of work that I am passionate about, computer science and wildlife.

On reflection, I started this project with an idea of what I wanted to achieve, however I had limited idea of how I could reach the goal. Once I had started the project the scale became apparent as I had to select every aspect of the development (without the brief of a marking scheme like every piece of coursework to date). This included the programming language, system, database, approach, design of every aspect and finish with a worthwhile conclusion in such as small space of time with limited help.

Despite studying computer science for the last 4 years I still naively believed that it would be simple to reach my goal without issues along the way. However, in reality I encountered challenges in almost every iteration of the project. I experienced issues using the API and was briefly unable to retrieve any data, effectively use tags, or set date parameters and that was the very first stage of the project. When attempting to develop a raster grid, my approach was poor and after a week of working towards a solution for the grid I determined it was not possible. This was due to my approach being to find a function (like Basemap) that would draw a grid for me. I certainly learnt that it is simpler and much more rewarding to sit down with a pen and paper and draw out an algorithm, than to rely on packages and built in functions, and it proved to be very possible!

I have certainly improved my technical skills during this project. Having never programmed before starting university, if you'd informed me that I would have implemented scripts that could extract geotags from social media, plot them onto a UK map, and compare them with ground truth data to detect trends and interesting behaviours of animal's species using a programming language like Python, I would not have believed it. I have improved my programming skills to the point where, if I thought of something project related that I wanted to research, I could believe I could certainly manipulate the scripts or create a new one to produce useful results.

I have enjoyed the wildlife related aspects of the project, such as researching animal species, learning about different migrations and hibernations, and analysing the results to suggest reasons for particular behaviours that have been uncovered. I have certainly learnt a lot and have further fuelled my interest in wildlife, and especially how technology can be utilised to benefit conservation.

11.0 Appendix

11.1 - Figure 1: FlickrDataCollection.py algorithm

```
var commonName = STRING(commonName)
var latinName = STRING(latinName)
search = Flickr.search(tag='commonName','latinName', place='uk')
var totalPages = int(search.pages)
FOR i in range(0,totalPages)
        var photos = Flickr.search(tag='commonName','latinName', place='uk', page='i')
        FOR j in range(0, photos.perpage)
                var photo_id = photos.id
                var photo_title = photos.title
                var photoInfo = Flickr.getInfo(photo_id=photo_id)
                var datatime = photoinfo.datetime
                var longitude = photinfo.longitude
                var latitude = photoinfo.latitude
                var URL = photoinfo.url

                INSERT INTO Table photo_id, photo_title, datetime, longitude, latitude, URL
        END FOR
END FOR
```

11.2 - Figure 2: Bar Chart Timeline Algorithm

```
var species = parameter(1)
var year = parameter(2)
var month = parameter(3)
FOR i in range(year, 2017):
        FOR j in range(month, 12):
                var count = SELECT Count(photo_id) from species where YEAR = year, MONTH = month)
                IF month == 1 or month == 2 or month == 12:
                        plot_winter = plt.bar(count)
                END IF
                IF month == 3 or month == 4 or month == 5:
                        plot_spring = plt.bar(count)
                END IF
                IF month == 6 or month == 7 or month == 8:
                        plot_summer = plt.bar(count)
                END IF
                IF month == 9 or month == 10 or month == 11:
                        plot_autumn = plt.bar(count)
                END IF
        END FOR
END FOR
```

11.3 - Figure 3: Line Graph Timeline Algorithm

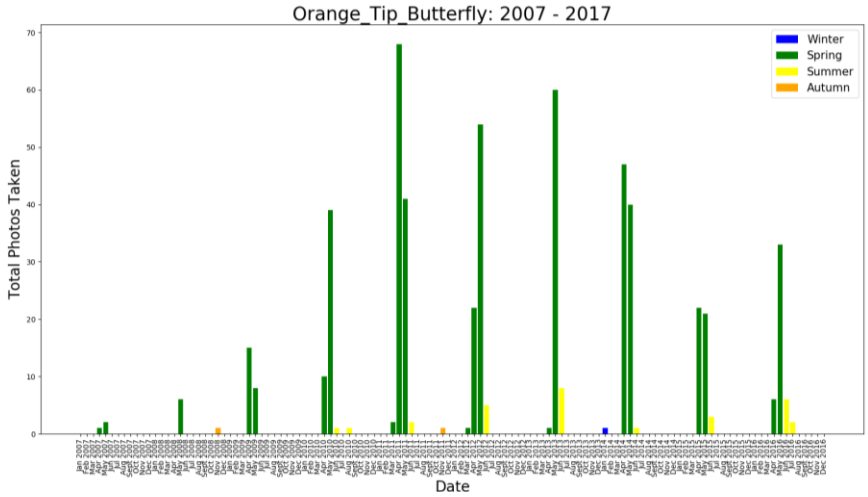
```
Var yearCounts= []
FOR i in range(year, 2017):
        FOR j in range(month, 13)
                Var count = SELECT Count(photo_id) from species where YEAR = year, MONTH = month)
                yearCount.append(count)
        END FOR
        plt.plot(yearCount)
END FOR
```

11.4 - Figure 4: Data Conversion Algorithm

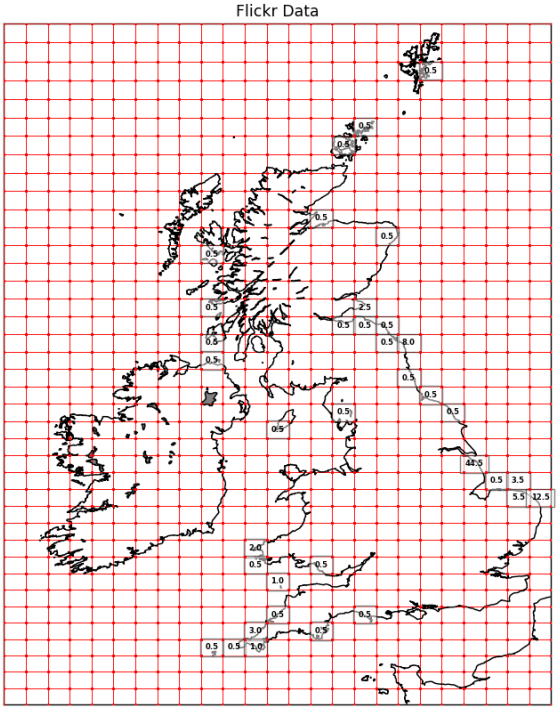
```
inputCSV = NBNData.csv
outputCSV = NBNLonLat.csv

FOR row in inputCSV:
        grid_ref = row[6]
        IF grid_ref[-1].isalpha() == True:
                grid_ref = grid_ref[:-1]
        IF grid_ref is irish:
                grid = irish2grid(grid_ref)
                E,N = grid2EN (grid)
                LL = irishEN2LL(E,N)
                Writerow(outputCSV)
        ELSE:
                grid = british2grid(grid_ref)
                E,N = grid2EN (grid)
                LL = britishEN2LL(E,N)
                Writerow(outputCSV)
        END IF
END FOR
```

11.5 - Figure 5: Orange Tip Butterfly Timeline



11.6 - Figure 6: Grey Seal Map Projection



12.0 References

1. Jones, C. (2016 Characteristics of Geographical Information and Spatial Data Models) Large Scale Databases Lecture, Slide 7.
2. LonLat.net. (2017) Latitude And Longitude Finder On Map Get Coordinates. Available at: http://www.latlong.net/ (Accessed: 20 April 2017).
3. Snorfalorpagus. (2014) Converting British National Grid and Irish Grid References: A Practical Example. Available at: https://snorfalorpagus.net/blog/2014/08/12/converting-british-national-grid-and-irish-grid-references-a-practical-example/ (Accessed: 20 April 2017)
4. Wikipedia. (2017) Irish grid reference system. Available at: https://en.wikipedia.org/wiki/Irish_grid_reference_system (Accessed: 20 April 2017)
5. Wikipedia. (2017) Citizen science. Available at: https://en.wikipedia.org/wiki/Citizen_science (Accessed: 20 April 2017)
6. Wikipedia. (2017) National Biodiversity Network. Available at: https://en.wikipedia.org/wiki/National_Biodiversity_Network (Accessed: 20 April 2017)
7. NBN. (2017) About the NBN Atlas. Available at: https://NBNatlas.org/about-NBN-atlas/ (Accessed: 20 April 2017)
8. RSPB. (2015) RSPB giving nature a home. Available at: https://ww2.rspb.org.uk (Accessed: 20 April 2017)
9. British Trust of Ornithology. (no date) About BTO. Available at: https://www.bto.org/about-bto (Accessed: 20 April 2017)
10. Eeles, P. (2016) About … UK Butterflies. Available at: http://www.ukbutterflies.co.uk/ukb.php (Accessed: 20 April 2017)
11. Omori, M. Hirota, M. Ishikawa, H. Yokoyama, S. (2014) Can Geo-tags on Flickr Draw Coastline?. SIGSPATIAL14 (Accessed: 20 April 2017)
12. Zhu, Y. Newsam, S. (2016) Spatio-Temporal Sentiment Hotspot Detection Using Geotagged Photos. SIGSPATIAL16 (Accessed: 20 April 2017)
13. Gilozzo, G. Pettorelli, N. Haklay, M. (2016) Using crowdsourced imagery to detect cultural ecosystem services: a case study in South Wales, UK. Resilience Alliance. Ecology and Society 21(3). (Accessed: 20 April 2017)
14. Hollenstein, L. Purves, R. (2010) Exploring place through user-generated content: Using Flickr tags to describe city cores. JOURNAL OF SPATIAL INFORMATION SCIENCE. doi:10.5311/JOSIS.2010.1.3 (Accessed: 20 April 2017)
15. Tenerelli, P. Demsar,U. Luque S. Crowdsourcing indicators for cultural ecosystem services: A geographically weighted approach for mountain landscapes. CrossMark. Ecolgoical Indicators 64 (2016) 237-248 (Accessed: 20 April 2017)
16. Woodland Trust. (no date) Find your next adventure. Available at: https://www.woodlandtrust.org.uk/ (Accessed: 2 April 2017)
17. Gil, P. (2017) What Exactly is 'Twitter'? What is 'Tweeting'?. Available at: https://www.lifewire.com/what-exactly-is-Twitter-2483331 (Accessed: 2 April 2017)
18. The Statistics Portal. (2017) Number of monthly active Twitter users worldwide from 1st quarter 2010 to 1st quarter 2017 (in millions). Available at: https://www.statista.com/statistics/282087/number-of-monthly-active-Twitter-users/ (Accessed: 5 April 2017)
19. Twitter Help Centre. (2017) FAQs about adding location to your Tweets. Available at: https://support.Twitter.com/articles/78525 (Accessed: 5 April 2017)

20. Wu, S. Boston, M. (2013) Twitter and Privacy: Nearly one-in-five Tweets divulge user location through geotagging or metadata. Available at: https://pressroom.usc.edu/Twitter-and-privacy-nearly-one-in-five-tweets-divulge-user-location-through-geotagging-or-metadata/ (Accessed: 5 April 2017)
21. Internet Live Stats. (2017) Twitter Usage Statistics. Available at: http://www.internetlivestats.com/Twitter-statistics/ (Accessed: 5 April 2017)
22. Twitter Developers. (2017) Twitter Developer Documentation. Available at: https://dev.Twitter.com/overview/API (Accessed: 5 April 2017)
23. Xu, W. (2015) Twitter API Tutorial. Available at: http://socialmedia-class.org/Twittertutorial.html (Accessed: 5 April 2017)
24. Moreau, E. (2017) What is Instagram, Anyway? Here's What Instagram Is All About and How People Are Using It. Available at: https://www.lifewire.com/what-is-Instagram-3486316 (Accessed: 5 April 2017)
25. Statista. (2017) Number of monthly active Instagram users from January 2014 to April 2017 (in millions). Available at: https://www.statista.com/statistics/253577/number-of-monthly-active-Instagram-users/ (Accessed: 6 April 2017)
26. Instagram Help Centre. (2017) How do I add a location before sharing my photo or video?. Available at: https://help.Instagram.com/408972995943225 (Accessed: 6 April 2017)
27. Karjalainen, J. (2015) What percentage of Instagram photos contain geolocation data?. Available at: https://www.quora.com/What-percentage-of-Instagram-photos-contain-geolocation-data (Accessed: 6 April 2017)
28. Instagram. (2017) API Endpoints. Available at: https://www.Instagram.com/developer/endpoints/ (Accessed: 6 April 2017)
29. Lowensohn, J. (2008) Newbie's guide to Flickr. Available at: https://www.cnet.com/uk/news/newbies-guide-to-Flickr/ (Accessed: 6 April 2017)
30. ponzu. (2010) [Tips] How to geotag photos (and why). You can set approximate location, too.. Available at: https://www.Flickr.com/groups/1473559@N21/discuss/72157625122844968/ (Accessed: 6 April 2017)
31. Catt, R. (2009) 100,000,000 geotagged photos (plus). Available at: http://code.Flickr.net/2009/02/04/100000000-geotagged-photos-plus/ (Accessed: 6 April 2017)
32. Flickr. (no date) The Flickr Developer Guide: API. Available at: https://www.Flickr.com/services/developer/API/ (Accessed: 6 April 2017)
33. Whitaker, J. (2014) Introduction. Available at: https://matplotlib.org/Basemap/users/intro.html (Accessed: 8 April 2017)
34. Cartopy. (2017) A library providing cartographic tools for Python. Available at: http://scitools.org.uk/cartopy/ (Accessed: 8 April 2017)
35. Whitaker, J. (2017) pyproj 1.9.5.1. Available at: https://pypi.Python.org/pypi/pyproj (Accessed: 8 April 2017)
36. Hunter, J. Dale, D. Firing, E. Droettboom, M . (2012) Matplotlib mplot3d toolkit. Available at: http://matplotlib.org/mpl_toolkits/mplot3d/ (Accessed: 8 April 2017)
37. Pip. (2016) Installation. Available at: https://Pip.pypa.io/en/stable/installing/ (Accessed: 9 April 2017)
38. Conda. (2017) Intro to Conda. Available at: https://Conda.io/docs/intro.html (Accessed: 9 April 2017)
39. Wikipedia. (2017) Kullback-Leibler divergence. Available at: https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence (Accessed: 12 April 2017)
40. Wikipedia. (2016) Hellinger distance. Available at: https://en.wikipedia.org/wiki/Hellinger_distance (Accessed: 12 April 2017)

41. Voline, C. (2004) The Earth Mover's Distance. Available at: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/RUBNER/emd.htm (Accessed: 12 April 2017)
42. Data School. (2014) Simple guide to confusion matrix terminology. Available at: http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/ (Accessed: 12 April 2017)
43. Mans, L. (2012) Hellinger distance for discrete probability distributions in Python. Available at: https://gist.github.com/larsmans/3116927 (Accessed: 12 April 2017)
44. Data School. (2014) Simple guide to confusion matrix terminology. Available at: http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/ (Accessed: 15 April 2017)
45. Math is Fun. (2016) Correlation. Available at: https://www.mathsisfun.com/data/correlation.html (Accessed: 20 April 2017)
46. Confusion Matrix. (no date) Confusion Matrix Online Calculator. Available at: http://onlineconfusionmatrix.com/ (Accessed: 20 April 2017)
47. Wikipedia. (2016) Decimal degrees. Available at: https://en.wikipedia.org/wiki/Decimal_degrees (Accessed: 20 April 2017)
48. Hunter, B. (no date) Batch Coordinate Converter. Available at: http://ww2.scenic-tours.co.uk/serve.php?t=WoNlbJvoVlhuJL5405objaa8jVO8atNuwZV (Accessed: 20 April 2017)
49. Anonymous. (2013) Assessing the 2013 Atlantic Puffin wreck. Available at: http://www.ceh.ac.uk/news-and-media/blogs/assessing-2013-atlantic-puffin-wreck (Accessed: 22 April 2017)
50. Corcoran, P. and Jones, C. (2016 *Spatio-Temporal Modeling of the Topology of Swarm Behavior with Persistence Landscapes*. ACM SIGSPATIAL 2016 International Conference on Advances in Geographic Information Systems.
51. Choochaisri, S. (2016) Recall vs Precision an alternate way to understand and remember recall and precision. Available at: https://hackercollider.com/articles/2016/06/03/recall-vs-precision/ (Accessed: 30 April 2017)
52. Penman, R. (2013) Converting UK Easting / Northing to Latitude / Longitude. Available at: https://webscraping.com/blog/Converting-UK-Easting-Northing-coordinates/ (Accessed: 30 April 2017)
53. Osborne, S. (2017) Albatross numbers on remote islands are being counted via space. Available at: http://www.independent.co.uk/news/science/albatross-space-numbers-count-space-satellites-chatham-islands-birds-northern-royal-digitalglobe-a7716806.html (Accessed: 04 May 2017)