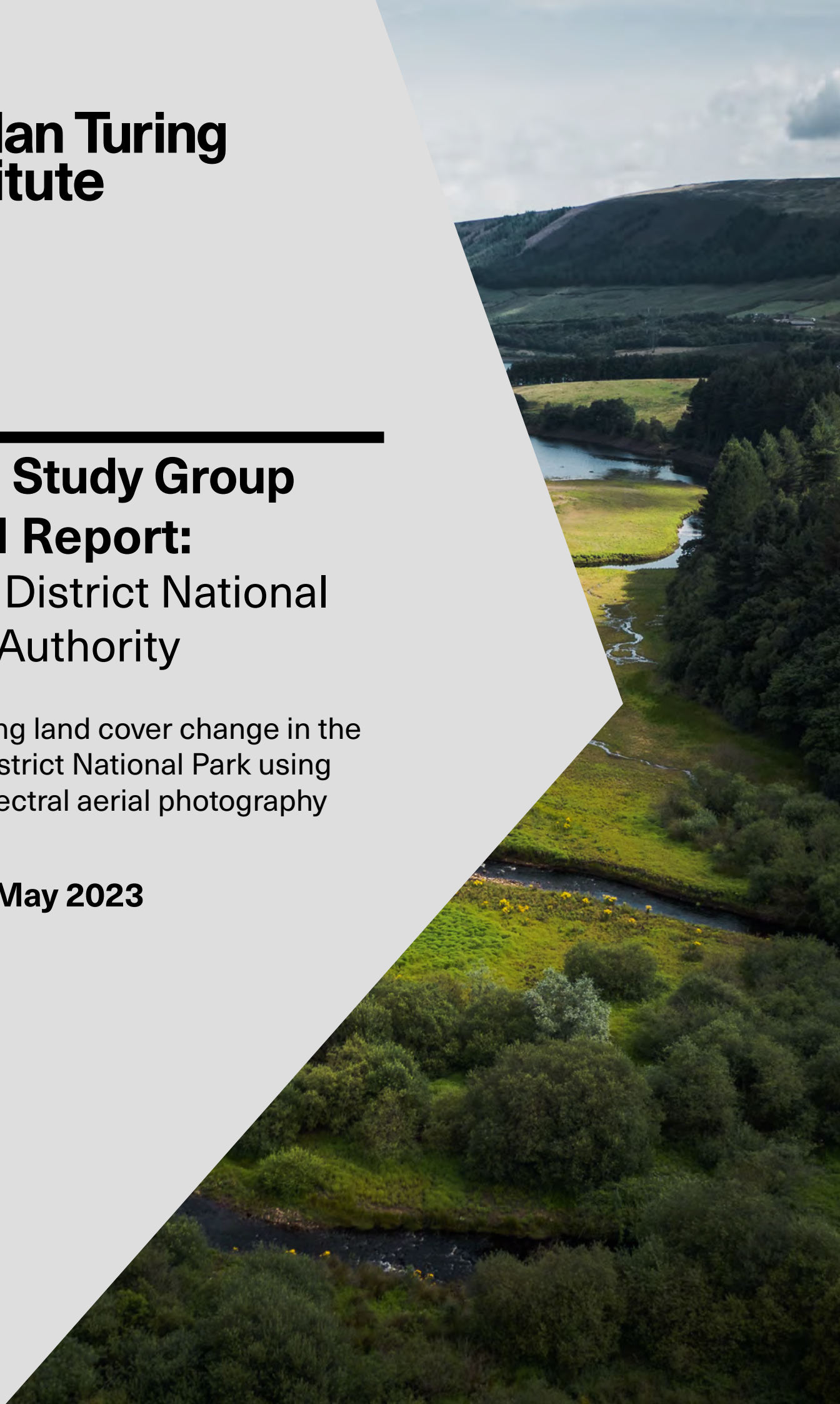
The Alan Turing Institute

Data Study Group Final Report:

Peak District National Park Authority

Detecting land cover change in the Peak District National Park using multispectral aerial photography

22-26 May 2023



DOI: 10.5281/zenodo.10090623

Contents

1 Challenge overview

The challenge set by the Peak District team involves making a *robust* and *scalable* algorithm to identify changes in the land cover, including changes in land use and habitats, using aerial photography images from two time periods roughly 10 years apart. This model should ideally be able to incorporate new data every 5 years and potentially be portable to application by other national parks in England.

The data consists of 12.5 cm ground resolution, ortho-rectified, geo-referenced true colour (RGB) files and 50 cm ground resolution, ortho-rectified, geo-referenced infrared (IR) files. The files represent 89 1 square-km areas chosen as a training set, representing a total area of 1,439 square-km of the park. In addition, there is secondary data, including files for a 2 m ground resolution Digital Surface Model and a 5 m ground resolution Digital Terrain Model.

The primary objectives for this challenge are:

- Develop *unsupervised* AI/ML algorithms for detecting change across The Peak District National Park that can be used to monitor changes in land use in future surveys and for other National parks.
- Produce change maps that detail regions where land use has been altered between 2010 and 2020.
- Identify the degree of changes in areas of conservation interest (i.e. peat restoration, deforestation, growth of bracken habitats and disappearance of dry stone walls).

An unsupervised approach that classifies the changes in land cover features between the two-time points is superior to applying the same supervised model on the two-time points and comparing the results, as erroneous predictions in the two models can compound into misleading change predictions. Re-labelling a new dataset is extremely time-consuming and represents a huge challenge.

Differences in land use will be impacted by the seasonal variation between the two acquisition dates. In addition, the impact of different acquisition times (which will elongate the shadows of various features) has not been considered.

2 Project Overview

2.1 Setting the Scene

The Peak District National Park was the first of Britain's 15 national parks and is one of the busiest national parks in the world. New technologies, climate change, more people and changing lifestyles mean that our potential to negatively change the environment and the appearance of the landscape is far greater now than in any previous generation. Through a lack of data we are struggling to understand these changes at a landscape scale and this makes it difficult to target our conservation efforts. Therefore, mapping the extent of land use and land cover (LC) categories is essential for better environmental monitoring, urban planning, nature protection, as well as long-term climate adaptation efforts. However, available data often does not provide the necessary (spatial and LC class) detail to support these use-cases. In fact, the last time land cover and land use were monitored (by manual annotation) to a high degree of accuracy and detail in UK National Parks was 1991. Consequently, in order to target small-scale and specific habitats, we urgently need to develop LC maps at high resolution and class detail, using automated methods to be able to efficiently scale to large areas of interest.

We have begun to address this problem in the past year. Using 12.5 cm aerial photography, we have deployed a multi-stage semantic segmentation approach to predict LC at a very high level of detail. This supervised-learning approach allows us to map the present-day LC with 71% to 97% accuracy, depending on the specific class.

The aim of the DSG is to develop a change-detection algorithm for land cover in the Peak District National Park. We have multispectral aerial imagery available from 2010 and 2020, and we would like to identify, map and understand the changes that have occurred in the past decade across the National Park.

Our work so far has been able to map the current land cover (LC) based on RGB aerial photography with a high degree of accuracy. However, we are also interested in mapping the change in LC that has occurred over the past decade. Applying our method to old imagery and comparing

predictions across years is sub-optimal, because erroneous predictions on either year would lead to false predictions of land cover change. Therefore, we would like to be able to infer LC change directly from the two dates of aerial photography.

Theoretically, the number of possible LC changes could scale quadratically with the number of LC classes (20) - although some LC changes are much more likely than others. Manually annotating these changes, even for the purpose of training machine learning models, is costly and very resource-intensive. Therefore, we would like to develop an unsupervised approach instead that detects changes in land cover from the raw image pixels directly and clusters changes across the National Park based on similarity. The product you develop during this DSG will be delivered to our ecologists, who can classify these inferred changes post hoc, using our resources efficiently by focusing on LC changes that stand out in the analysis.

Some changes that we know have occurred but are unsure of their degree of change and distribution across the Park are:

- Growth and expansion of some species, such as bracken, which, when unmanaged, can hinder the growth of other (more favourable) vegetation such as trees.
- Conversion from monotonous habitats into mosaics and vice versa (e.g., between grass and heather).
- Responses to land management changes include plantation woodland being converted to broadleaved woodland or rough grassland being intensified.
- Deforestation and the response of trees to diseases such as Ash DieBack, particularly at small scales that don't meet the Forestry Commission's criteria for woodlands ($>$0.5 ha in size).
- Restoration of eroded peat (due to the Industrial Revolution) to healthy peat bogs with vegetation.
- Disappearance of dry-stone walls, a key cultural heritage feature of the Peak District, as well as hedges

Being able to map these changes would be an incredibly powerful tool to

guide conservation work in the Peak District.

2.2 Objectives

- Develop *unsupervised* AI/ML algorithms for detecting change across The Peak District National Park that can be used to monitor changes in land use in future surveys and for other National parks.
- Produce change maps that detail regions where land use has been altered between 2010 and 2020.
- Identify the degree of changes in areas of conservation interest (i.e. peat restoration, deforestation, growth of bracken habitats and disappearance of dry stone walls).

3 Data overview

For this challenge 89 tiles covering a random subset of the Peak District National Park have been made available (Figure 1a). This high-quality remote sensing data has been captured by Airbus and Blue Sky. The Peak District team have an APGB license to use the data. These tiles are 1km x 1km in size. The corresponding files for each data type is identified with a 6-digit grid-reference number (SXNNNN) in the filename.

The data consists of 12.5 cm ground resolution, ortho-rectified, geo-referenced true colour (RGB) files and 50 cm ground resolution, ortho-rectified, geo-referenced infrared (IR) files (Table 1). The files represent 89 1 km^2 areas chosen as a training set, representing a total area of 1,439 km^2 of the park. In addition, there is secondary data including files for a 2 m ground resolution Digital Surface Model and a 5 m ground resolution Digital Terrain Model. The digital surface and terrain models have been captured for 2020 only.

3.1 Dataset description

Data in this project can be split into 4 main categories. These categories are detailed in the table below:

(a) (b)

Figure 1: Spatial distribution of sample tiles and aerial image dates (Aerial Photography Great Britain) within the Peak District National Park UK

Data Description		
Data Type	Resolution (m)	Description
Aerial Photos	12.5	Red, Green and Blue bands
Colour Infrared	50	Infrared band
Digital Terrain Model	5	Captures the underlying Earth
Digital Surface Model	2	Captures artificial and natural features in the environment

Table 1: Summary of available datasets .

The land use schema from the Natural England survey conducted in 1991 was used for the labelled model in 2020 (Figure 2). This was a result of a manual survey of changes in aerial photographic images between the 1970s and 1980s.

The colour-infrared (CIR) aerial photographs are taken from the near-infrared portion of the spectrum with wavelengths between 700 nm to about 900 nm, and the data includes values for 3 channels that can be translated to RGB for viewing the images (so called *false-colour* images). The IR data is more descriptive in vegetation classes than the RGB data and signal has an indirect correlation with biomass. Live vegetation will

LC80-main	LC80	LC20	Name	New?
C (Wood and forest land)	C1	C1	Broadleaved high forest	-
C	C2	C2	Coniferous high forest	-
C	C3	C3	Mixed high forest	-
C	C4 (Scrub)	C4a	Scrub	Yes
C	C4	C4b	Scrub pasture	Yes
C	C1-C4	C4c	Woodland/scrub edge	Yes
C	C5	C5	Clear felled/newly planted trees	-
D (Moor and heath land)	D1	D1	Upland heath	-
D	D2b	D2b	Upland grass moor	-
D	D2d	D2d	Blanket peat grass moor	-
D	D3	D3	Bracken	..
D	D6a	D6a	Upland heath/grass mosaic	-
D	D6b	D6b	Upland heath/bracken mosaic	-
D	D6c	D6c	Upland heath/blanket peat mosaic	-
D	D7a	D7a	Eroded bare peat	-
D	D7b	D7b	Eroded mineral soils	-
E (Agro-pastoral land)	E1	E1	Cultivated land	-
E	E2a	E2a	Improved pasture	-
E	E2b	E2b	Rough pasture	-
F (Water and Wetland)	F2	F2	Open water, inland	-
F	F3a	F3a	Peat bog	-
F	D2/E2	F3d	Wet grassland and rush pasture	Yes
G (Rock and coastal land)	G2	G2	Inland bare rock	-
H (Developed land)	H1a	H1a	Urban area	-
H	H1b	H1b	Major transport route	-
H	-	H1c	Minor transport route	Yes
H	H1a	H1d	Urban greenspace	Yes
H	H2a	H2a	Quarries and mineral working	-
H	H2b	H2b	Derelict land	-
H	H3a	H3a	Isolated farmsteads	-
H	H3b	H3b	Other developed land	-
I (Unclassified land)	I	I	Unclassified land	-

Figure 2: 1991 land use classification schema from land survey Natural England

Table 2: Selected reference tiles.

Tiles of Interest		
Ordnance Survey Reference	Shape Number	Features Changed
SK0987 (Fig. 3)	63	Bare peat to vegetation
SK1091 (Fig. 4)	66	Deforested patches of woodland
SK2096 (Fig. 5)	47	Bracken growth on hills
SK0961 (Fig. 6)	25	Dry stone wall destruction

appear very red and the colour intensity fades as the plant vigour decreases. Some classes (especially deciduous plants) will change appearance over the calendar year as they absorb light differently in different seasons. Bare soils will appear white, blue or green with darker colours indicating more moisture. Changes in man-made structures will be less apparent in the IR images than the RGB data and the colours will depend on the material. Water will appear in blue or black shades except for shallow features which will reflect the material of the bottom of the feature, like a stream bed, for example.

Initial data exploration was conducted using QGIS software package and python scripts which were used to extract and visualise files from specific grid reference numbers to compare different years side-by-side, as well as the relevant terrain and elevation images. This allowed identification of key changes in the land use that can be used to validate the models tested.

Early on in the project there were discussions with the project owners to identify tiles which showed clear changes between 2010 and 2020 in the areas of conservation interest. These would be tiles that we could use as positive tests of land use change with the various approaches. The four main tiles that were identified are detailed in 2.

3.2 Data quality issues

One of the primary issues that could impact how the downstream techniques function is the seasonality in the tiles. Some of the tiles captured in 2010 and 2020 have been captured in different months of the year. The data was collected from the middle of UK spring until late

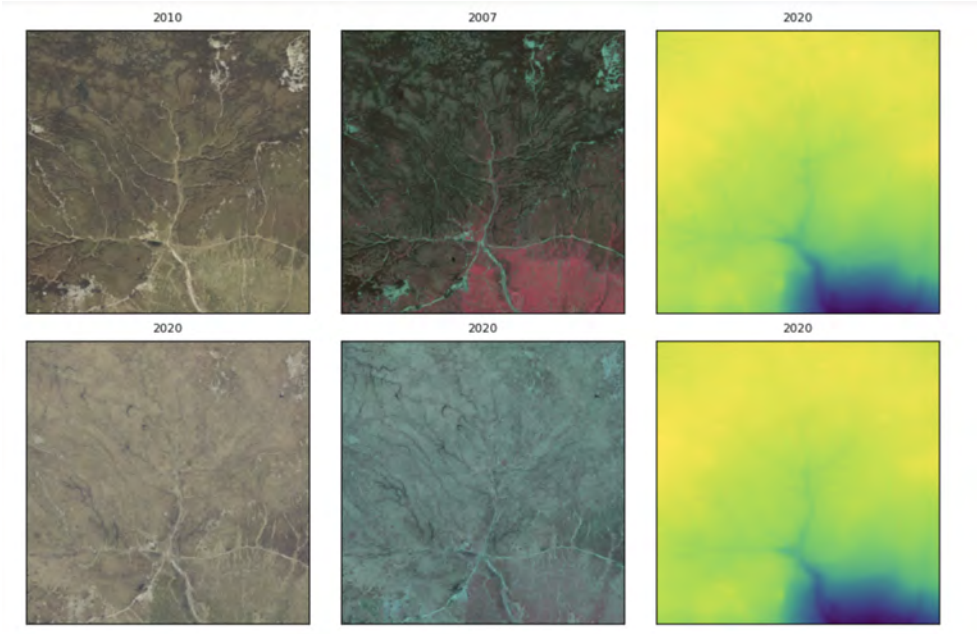


Figure 3: Bare peat to vegetation (SK0987) © Bluesky International Limited and Getmapping Plc 2023

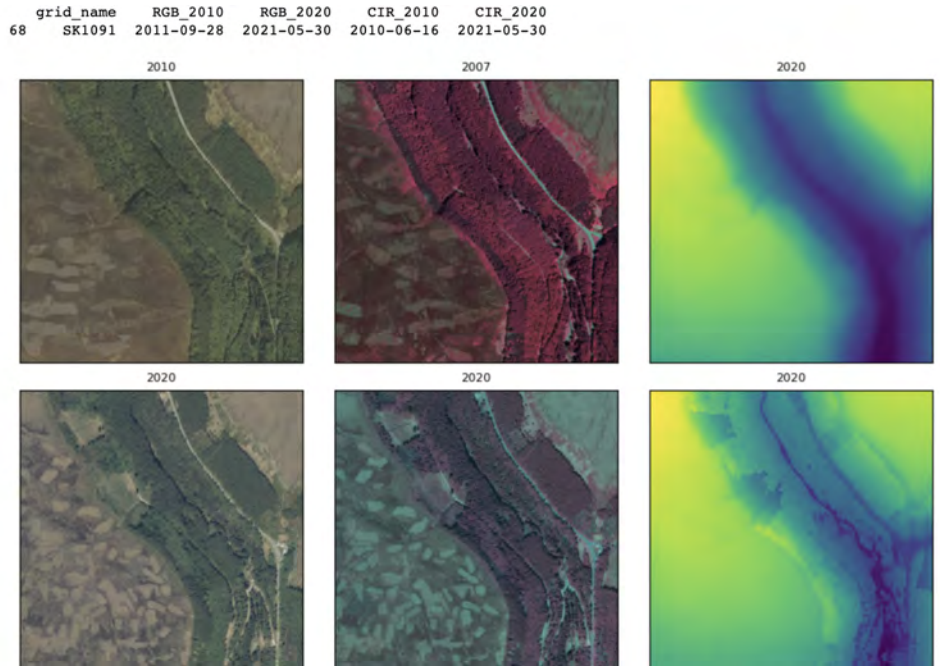


Figure 4: Deforested patches of woodland (SK1091) © Bluesky International Limited and Getmapping Plc 2023

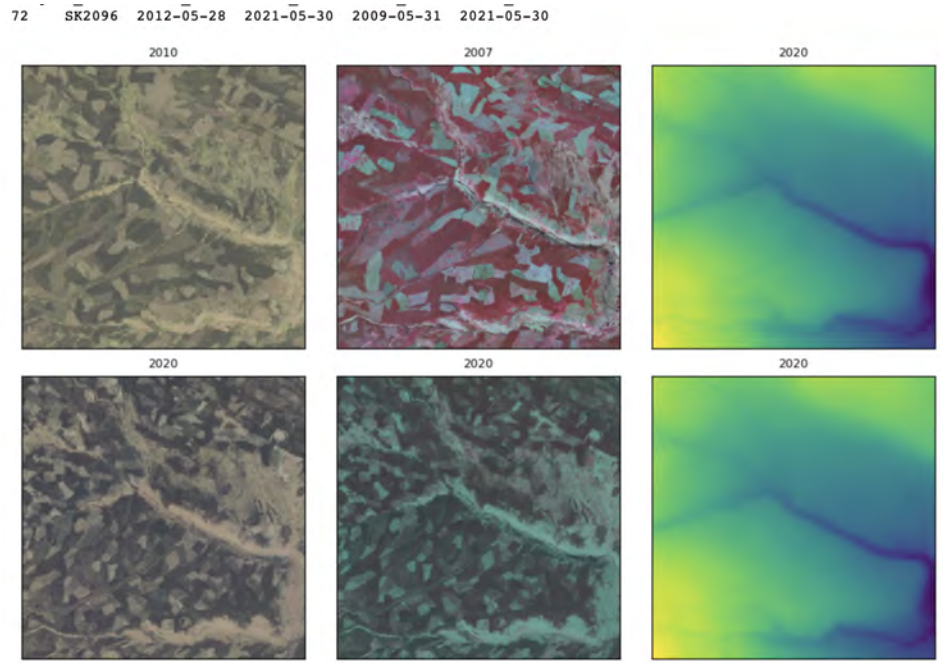


Figure 5: Bracken growth on hills (SK2096) © Bluesky International Limited and Getmapping Plc 2023

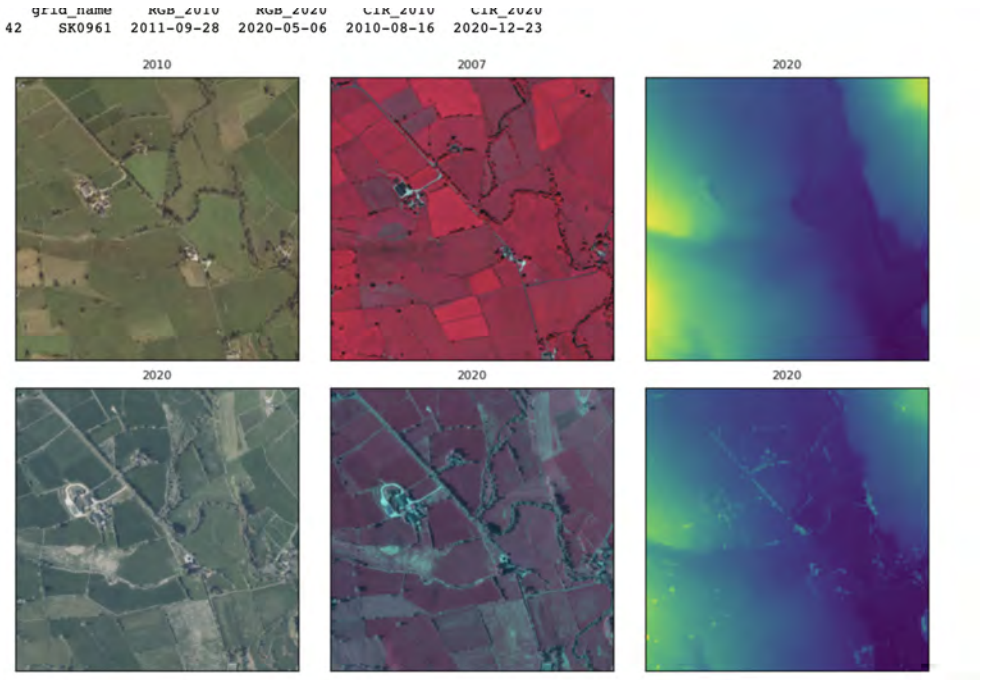


Figure 6: Dry stone wall destruction (SK0961) © Bluesky International Limited and Getmapping Plc 2023

summer. This will have an impact on the signatures from the various vegetation since their shape/colour/size can be different throughout the seasons. For example, trees in a forested area that are in leaf in one dataset versus bare in another, or fields that are in flower/green/mown can show up as a changes that aren't relevant to land use change. We have extracted the capture dates from the relevant metadata files for both RGB and IR data to evaluate whether or not seasonal variation is affecting model performance. The original spreadsheet provided with the data had incorrect capture dates for the 2020 IR files.

It is also noted that the time period between the datasets is not consistent for matched grid areas with some being several years more or less than 10 years apart.

The infrared data seems to have a uniformly more 'red' signature in the 2010 dataset than the 2020 based on qualitative assessment and histograms from selected tiles (Figure 7). This seems independent of seasonal differences and may need to be normalised before use in the models, and/or standardised in one of the layers to avoid training on normalised data.

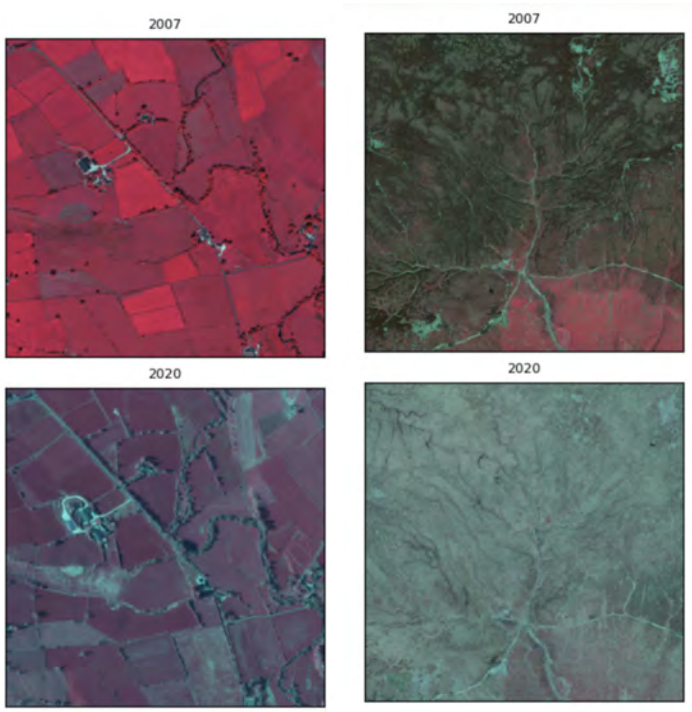


Figure 7: IR data from (top) 2007 contains more red colour than (bottom) the 2020 time point.

4 Change detection in remote sensing images

The objectives of the Peak District National Park's data challenge are closely related to the field of change detection for remote sensing applications. It encompasses the task of capturing spatial and temporal changes in aerial images. A variety of methods of different levels of sophistication have been proposed for this task [1]. The output of the majority of these algorithms are change maps. Given two input images representing the same area at different points in time, a change map will show some sort of notion of change measure between the inputs assigned to each pixel pair. This takes the form of an array of the same dimension as the input images with values either representing the probability of change having occurred, or a binary classification of change/no change at each point in the grid. We considered a variety of statistical and machine learning based methods (supervised and unsupervised), which are described in detail in section 5.

5 Techniques investigated

5.1 Iteratively Reweighted Multivariate Alteration Detection (IR-MAD)

IR-MAD is a widely used method for land cover change detection that leverages multivariate statistical techniques. It aims to identify significant changes between two or more multivariate datasets, such as satellite images, by assigning weights to the variables based on their contribution to the change detection process.

Let's consider two datasets: $\mathbf{X}_1$ and $\mathbf{X}_2$, where $\mathbf{X}_1 = [\mathbf{x}_1^1, \mathbf{x}_2^1, \ldots, \mathbf{x}_n^1]$ and $\mathbf{X}_2 = [\mathbf{x}_1^2, \mathbf{x}_2^2, \ldots, \mathbf{x}_n^2]$ are p-dimensional matrices representing the n samples (pixels) in each dataset.

The goal is to detect changes between the two datasets by calculating an alteration vector $\mathbf{D}$, where each element d_i represents the alteration of the i-th variable.

IRMAD calculates the alteration vector as follows:

$$\mathbf{D} = \frac{\sum_{i=1}^{n} w_i(\mathbf{x}_i^1 - \mathbf{x}_i^2)}{\sum_{i=1}^{n} w_i}, \tag{1}$$

where w_i is the weight assigned to the i-th variable.

The weights w_i are computed iteratively using a robust statistical measure, such as the median absolute deviation (MAD), to downweight the influence of outliers in the change detection process. The iterative procedure is as follows:

1. Initialize the weights w_i to equal weights (e.g., $w_i = 1$ for all i).
2. Compute the alteration vector $\mathbf{D}$ using Equation (1).
3. Compute the robust scale measure, for example, the MAD of $\mathbf{D}$, denoted as S.
4. Update the weights w_i using a weight function f based on the robust scale measure S:

$$w_i = f\left(\frac{d_i}{c \cdot S}\right), \tag{2}$$

   where d_i is the i-th element of $\mathbf{D}$, c is a tuning parameter, and f is a function that assigns weights based on the scaled alteration values.
5. Repeat steps 2-4 until convergence or a predefined number of iterations is reached.

The final alteration vector $\mathbf{D}$ obtained from the iterative process represents the detected changes between the two datasets. By examining the magnitudes and patterns of the elements in $\mathbf{D}$, one can identify the significant changes in land cover.

The change map generated by IR-MAD for one of the areas of conservation interest where deforestation is known to occur is an interesting case study for the success and limitations of the method as it was implemented here. In Figure 8, the negative image of the change map is overlaid with the predictions from the 2020 classification. The

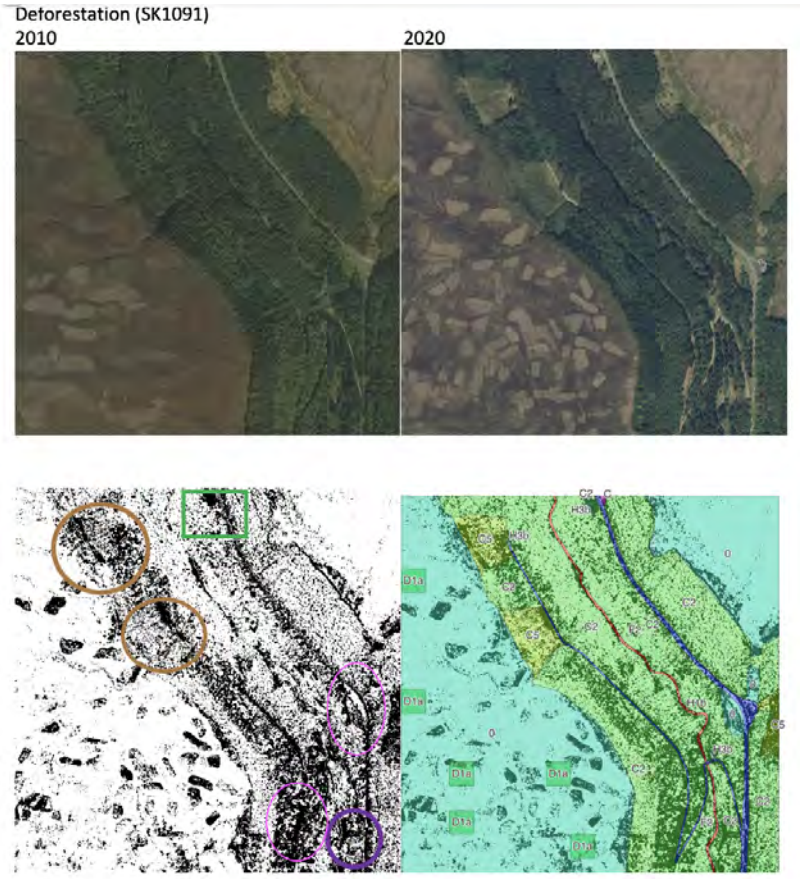


Figure 8: Deforestation (top) aerial photography tile SK1091 with (bottom left) change map with a binary scale, white pixels are changed and black pixels are unchanged, (bottom right) changes classified using the 2020 model predictions of landcover.

results of the 2020 automated machine learning classification model highlights the brown areas on the land use survey ('C5') which represent areas of recently felled or planted trees. The change map highlights these areas as areas of change between the two time studies (brown circles). Another interesting area of change is the farmland in the very top centre of the grid ('H3b') which could represent expanded farming use (green square). There are several areas along the riverbed (pink circles and roads (purple circle) that also show change–perhaps more tree felling. The clusters of changed pixels inside the large grassland section (D1a) probably represent blooming plants that change seasonally. This is an area where the approach needs fine-tuning to extract signal from noise.

The regeneration of peat bog reflects an increase in vegetation between the timepoints and is clearly reflected in the change map (Figure 9. Large-scale restoration of bog peat habitats have taken place in the Peak District National Park to increase carbon and water capture and return valuable biodiversity. The IR-MAD results have identified areas in the top right corner of the images that may be a result of this restoration. The change maps have also identified two regions (red circles) that are identified as 'bare rock' (G2) in the automated feature classification. Does this represent a loss of vegetation?

5.2 Change Vector Analysis

Change Vector Analysis (CVA) is a popular method used to detect land cover changes by analyzing the changes in spectral values between two or more time periods. It quantifies the magnitude and direction of changes in the multi-dimensional feature space. The CVA method can be applied to satellite imagery or other remotely sensed data.

Let's consider two time periods, t_1 and t_2, and assume that we have a set of N spectral bands representing the land cover information for each time period. For each pixel in the image, the spectral values at time t_1 can be denoted as $\mathbf{X}_{t_1} = [X_{t_1,1}, X_{t_1,2}, \ldots, X_{t_1,N}]$, and the spectral values at time t_2 as $\mathbf{X}_{t_2} = [X_{t_2,1}, X_{t_2,2}, \ldots, X_{t_2,N}]$.

The change vector, $\mathbf{V}$, is computed as the difference between the two spectral vectors:

Peat regeneration (SK0987)

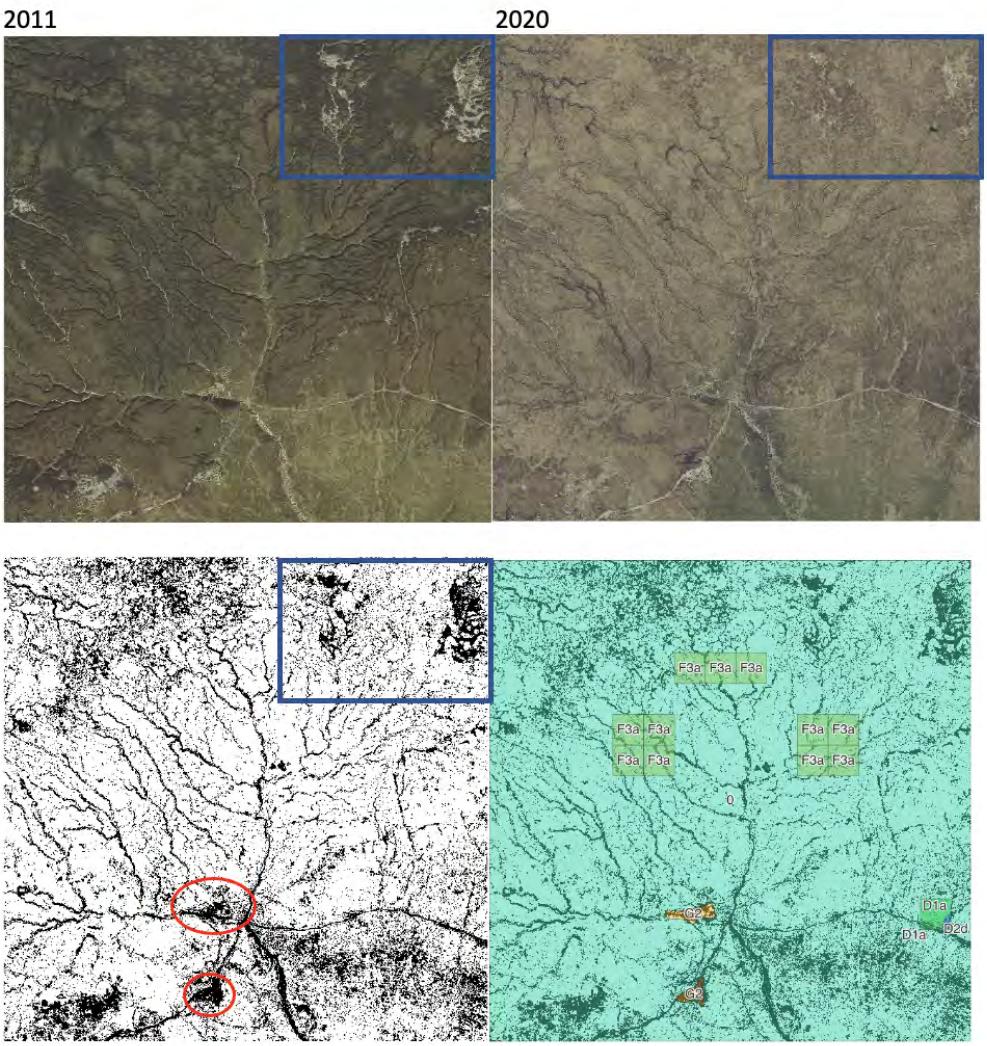


Figure 9: Caption

$$\mathbf{V} = \mathbf{X}_{t_2} - \mathbf{X}_{t_1}$$

The magnitude of the change vector, $|\mathbf{V}|$, represents the overall change in the pixel's spectral values. It can be calculated using the Euclidean distance:

$$|\mathbf{V}| = \sqrt{\sum_{i=1}^{N} (X_{t_2,i} - X_{t_1,i})^2}$$

The direction of the change vector can be determined using the angle between $\mathbf{V}$ and a reference vector, $\mathbf{R}$. The reference vector is typically chosen to represent the expected or natural changes in the feature space, such as the annual variation in vegetation. It is important to select an appropriate reference vector for accurate change detection.

The angle, θ, between the change vector and the reference vector can be computed using the dot product:

$$\theta = \arccos\left(\frac{\mathbf{V} \cdot \mathbf{R}}{|\mathbf{V}| \cdot |\mathbf{R}|}\right)$$

where $\cdot$ denotes the dot product and $|\cdot|$ represents the magnitude of a vector.

By analyzing the change vectors for each pixel in the image, it is possible to classify the land cover changes into different categories based on their magnitudes and directions. Thresholds can be set to determine the significance of changes and classify them as stable or changed areas.

The CVA method provides valuable insights into land cover changes, allowing researchers and decision-makers to monitor and manage environmental resources effectively.

5.3 Slow Feature Analysis

Slow Feature Analysis (SFA) is a method used for extracting slow-varying features from time series data. It can be employed for detecting land cover change by analyzing temporal patterns in satellite imagery. SFA aims to identify changes that occur slowly over time, which are often indicative of land cover transitions.

The SFA algorithm operates in several steps:

1. **Preprocessing:** Convert the satellite imagery into a time series dataset. Each pixel location is treated as a separate time series, representing the spectral values over time.
2. **Normalization:** Normalize the time series data to have zero mean and unit variance. This step helps in removing any global trends or biases in the data.
3. **Temporal Derivatives:** Compute the temporal derivatives of the time series data. This involves estimating the rate of change for each pixel location.
4. **Decorrelation:** Decorrelate the temporal derivatives by applying a linear transformation. The transformation ensures that the resulting features are statistically independent.
5. **Slowness Principle:** Identify the features that change slowly over time. The slowness principle assumes that important changes in the land cover typically occur slowly, and thus, the slow-varying features capture these changes.
6. **Inverse Transformation:** Reverse the linear transformation applied in the decorrelation step to obtain the slow-varying features in the original space.

To detect land cover change, the SFA method primarily focuses on the extraction of slow-varying features. These features are obtained by solving an optimization problem based on the slowness principle. The slowness measure is defined as the squared deviation between the temporal derivative of a feature and its delayed version. Mathematically, the slowness measure is given by:

$$\text{Slowness Measure} = \frac{1}{2}\sum_{t=1}^{T}\left(\frac{d\mathbf{y}_i(t)}{dt} - \lambda \mathbf{y}_i(t-1)\right)^2$$

where $\mathbf{y}_i(t)$ represents the i-th feature at time t, $\frac{d\mathbf{y}_i(t)}{dt}$ denotes its temporal derivative, and λ is a hyperparameter controlling the delay.

The optimization problem aims to find the feature $\mathbf{y}_i(t)$ that minimizes the slowness measure, subject to the constraint that the feature has unit variance. This can be expressed as:

$$\min_{\mathbf{y}_i} \frac{1}{2}\sum_{t=1}^{T}\left(\frac{d\mathbf{y}_i(t)}{dt} - \lambda \mathbf{y}_i(t-1)\right)^2 \quad \text{subject to} \quad \frac{1}{T}\sum_{t=1}^{T}\mathbf{y}_i(t)^2 = 1$$

Solving this optimization problem yields the slow-varying feature $\mathbf{y}_i(t)$, which can be further used for land cover change detection.

5.4 Clustering cropped images

Clustering is an unsupervised machine learning technique used to group similar data points together based on their inherent characteristics or similarities. It is employed in a wide range of domains such as data analysis, pattern recognition, image processing, and customer segmentation. The primary goal of clustering is to identify natural groupings within a dataset without any prior knowledge of the group labels. By assigning data points to clusters based on their similarity, clustering algorithms reveal hidden patterns or structures in the data. The process involves representing each item or observation in the dataset as a data point, which is characterized by measurable attributes known as features. These features determine the similarity or dissimilarity between data points, typically measured using a distance metric. Clustering algorithms, such as the popular k-means clustering algorithm, aim to partition the dataset into a predetermined number of clusters, denoted as 'k'. The k-means algorithm iteratively assigns data points to the nearest cluster centroid and updates the centroid based on the newly assigned points. This process continues until convergence, where the cluster assignments and centroids remain unchanged. K-means clustering is

Figure 10: Out of 400 tiles we have selected 10 representative tiles using the K-means clustering approach. This shows the 10 spatial location from the original 2010 SK0961 picture. The selected patches show the hot spots for changes to happen

widely used due to its simplicity and efficiency. Cluster validation techniques can be employed to evaluate the quality and validity of the clusters generated by the algorithm.

We believe that these patches can be used to produce synthetic data fro the future modelling as they tend to show some of the changes that have happened at the same location. So, instead of using the random overlapping of images, these patches can be used to show some of the actual changes that have happened over the years.

In future work, we would use k-means clustering with k as the number of classes in the dataset. If we do this for the old and the new data then we can have an indication of when one data point moves into or near another centroid indicating that it might have changed class.

5.5 Visualising features extracted from cropped in sections of images

Fig. 14 shows a visualization of features extracted from crops of the SK1091 images. We first crop the image from 8000 by 8000 pixels into 400, 400 by 400 crops. We can then use a pre-trained feature extractor to

Figure 11: 2020SK 10 representative images from the same location.

take the image into a representation space. We can then visualize this space in two dimensions using T-SNE in Fig. 13. Each of the blue and orange markers in the scatter plot in the figure represents one cropped image section, with blue showing the 2010 data and orange showing the 2020 data. Where a point does not overlap the point corresponding to the same image crop in old vs new, there has been a change in the feature representation and base image, with more distance indicating more change. We have added annotations showing the index of each of the crops, 0 to 400. We also annotate and show a few of the representative examples images where there is a larger gap between the two-time points. We notice that index 177 has picked the treeline change in SK1091. The changes in the upper right corner seem to relate the farmland.

In future work we can add additional information to this plot, with an example contained in the figure, by adding colors relating to the largest prominence class contained in the cropped image, this might show some clustering relating to the classes. This could help indicate if a cropped image has changed class, by seeing if any of the crops have moved towards a regen of a different color/class.

In the future, we can also add all of the images together into the analysis for a plot showing the changes in the entire dataset, rather than just the SK1091 image used in our analysis.

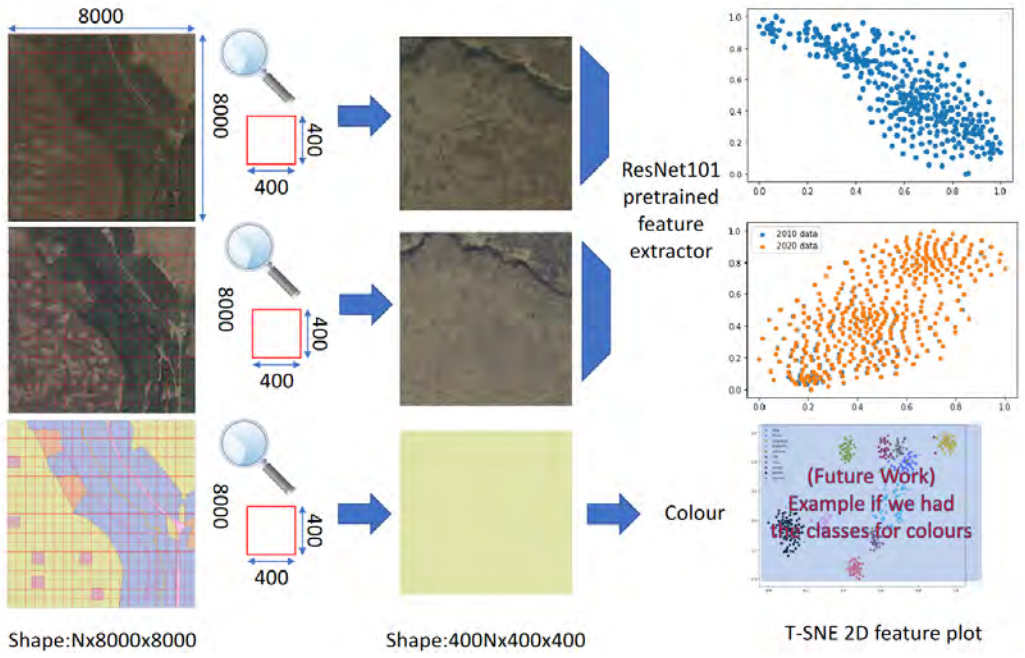


Figure 12: Methodology used to extract features from crops of images that are used as data points in a 2D distance plot. In future work, we can color the 2020 data points as in the example by the label data we did not have time to integrate.

Another possible improvement for this method would be to use the features extracted by an autoencoder trained on the peak district data, as these would be more relevant for the task than those of an ImageNet-trained model. For the analysis we can then use Central Kernel Alignment [9], which is the state-of-the-art for computing distances between extracted feature maps.

5.6 Change detection using siamese networks

Siamese neural networks (SNNs) are a neural network architecture commonly used for change detection in image data. They consist of multiple sister networks with shared weights. Each sister network produces embedding vectors for its own set of inputs. In supervised similarity learning, these networks are trained to enhance the distinction

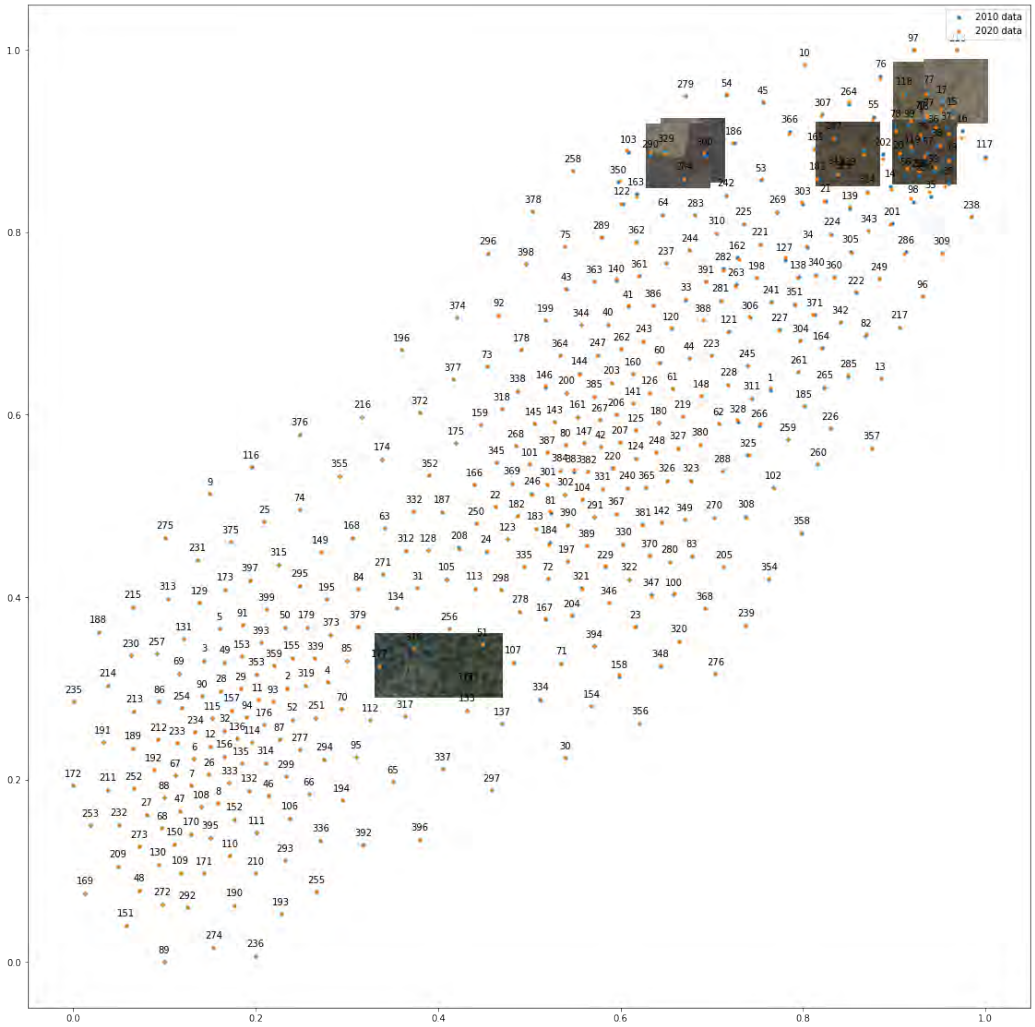


Figure 13: The SK1091 T-SNE plot of features from a single image split into 400, 400 by 400 pixel crops. The difference can be seen between the orange and blue scatter points.

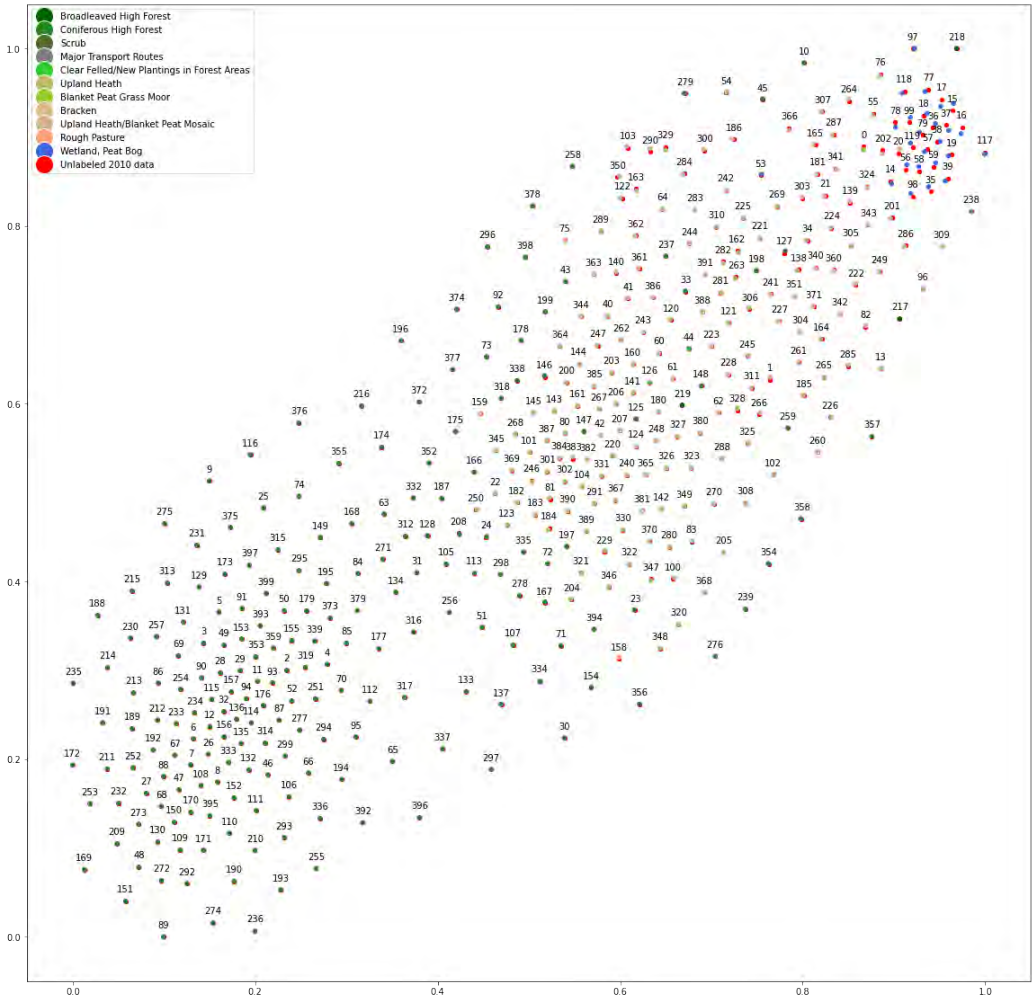


Figure 14: We have added the colors relating to the classes which have been labeled for the 2020 data. Where red is visible, that shows that the representation has moved and there has been change. The number annotations relate to the index of the crop, which can be used to view the crop and assess change.

(distance) between embedding vectors of inputs belonging to different classes. At the same time, SNNs strive to minimise the distance between embedding vectors of inputs belonging to similar classes. This training process creates embedding spaces that effectively represent the classification boundaries of the training inputs.

SNNs provide an attractive tool for building change detection systems and have been successfully deployed in a wide range of real-world applications (facial verification, product recommendation etc.). However, SNNs in their original form require supervised learning. The dataset must contain labels indicating change or no change, which are not available for our data. A number of modifications of the original SNN architecture, suitable for unsupervised learning have been proposed in recent literature. As part of the technical exploration, we reviewed a number of papers claiming to use unsupervised learning to detect change in aerial photography or satellite imagery. We were particularly interested in those reports that made code available and so focused on those listed in the Change Detection Repository and written in Python. Candidate solutions considered included [18, 14, 13, 12, 6, 5].

Although code for many of these examples was available, the major obstacle to employing most network architectures was the lack (or complete absence) of documentation to guide development. An additional hurdle was that the code available frequently used outdated libraries that required extra steps to build virtual environments (e.g. TensorFlow v.1.9 / Python 3.6) or refactor for use with the environment available on virtual machines. Although such obstacles are surmountable, it was not feasible to spend the additional time doing so within the limited window of the Data Study Group (DSG). We also noted a lack of pretrained networks, which we hoped to adapt to test with images from the peak district, and could not quickly identify links between reported systems and commonly used model repositories (e.g. HuggingFace).

The constraints of available unsupervised systems emphasized both the value of developing image differencing pipelines that do not require training (Section **??**) and of developing synthetic data for training simpler (but better documented) supervised Siamese networks (see Section 5.7). More broadly, the state of the literature and the difficulty in replicating modelling work in the field emphasised the importance of creating

well-documented and reusable code that appears to be lacking in this area of the field. The robustness of a change detection system (either produced in the DSG, or more generally) thus depends not only on its performance across diverse image sets, but on its reproducible and user-friendly code base if it is to be used on a regular basis (i.e. to support change detection for future flights) or in conjunction other organisations within the national parks authority (i.e. to study change in other parks).

Given the importance of time in generating useful outputs and the importance of usable, well documented and modern code, we choose to move forward with network architectures described in sections 5.7.1, 5.9.1).

5.6.1 KPCA-MNet

KPCA-MNet is a method of computing binary and multi-class change maps from remote sensing image pairs [15]. It utilises Kernel Principal Component Analysis convolution to extract nonlinear spatial–spectral features in an implicit manner using randomly selected patches as training samples, performing what can be expressed as a convolution. These transformations can then be stacked as layers to construct a deep siamese KPCA convolutional mapping network. The resulting feature maps for two inputs are then subtracted and processed into a binary feature map. We adapted the code to work on examples from our dataset, producing binary change maps. However, we considered this method as a variation of the CVA and IRMAD methods. This is because even though it is inspired by SNNs, it does not allow us to learn from the entire dataset.

5.7 Synthetic change detection

This approach turns the given problem into a supervised, change map generation. Given a spatial location, we would like to be able to get two areal images from different moments in time and be able to output a "change map". We would like the change map to indicate the pixels that correspond to a land cover change between the two time points.

The data set for this problem will consist of a source image x, an image $\tilde{x}$ covering the same spatial area as the source image but which is taken at a different time step, and a change map m. However, we only have access to land cover annotations for the 2020 data so a change map cannot be computed. We can instead create an artificial data set. Given a source image x, we artificially create a modified version $\tilde{x}$, where we know exactly which pixels have been modified. This type of image modification is commonly used as a form of augmentation and it is commonly known as mixed-sample data augmentation. Examples of mixed-sample data augmentation include MixUp [17] and CutMix [16]. Mixup linearly interpolates two images, while CutMix crops a rectangular patch from an image and overlaps it on top of another one. The change map we generate will have the same dimenionality as the x and $\tilde{x}$ and each pixel will represent the probability of land cover change. That is, each pixel of the cover map will have a value between 0 and 1.

Data used

Since this method makes use of class information, we used a different data subset to the one made available in the main challenge. While there exists an overlap between the two, the land cover information is only available for a very restricted number of tiles. For processing this data, we used the public repository created by Thijs van der Plas.

Modifying images

For computational reasons, we choose to randomly crop patches of size 256×256 which we then rescale to 128×128 using biliniear interpolation.

When modifying images, it is important to ensure that we are not introducing artefacts that bias the model. For example, using rectangular patches to distort the source image would make the model learn to look for rectangular areas of change. We therefore want to ensure that sufficient variety in the distortion shape and granularity is present.

We also want to ensure that when creating the change map we take into account the case when we are mixing patches belonging to the same land cover class. For example, if we add an artificial forest patch inside a

forest image, then the change map should indicate that no change has occurred, since the type of land cover is actually preserved. For this reason, we used the data annotations provided. Note that it is possible to create such a synthetic data set without access to the ground truth but the quality of the trained model is expected to decrease significantly. We therefore did not explore this approach and chose to include the land cover information.

For creating change maps we use random mask generators as it is done in mask mixed-data augmentation. We then modify the mask to account for the land cover information. See Figure 15 shows examples of synthetic images we obtain.

The first modification method we use is based on the FMix [7] augmentation, where a mask is sampled from Fourier space. The obtained mask has more naturally edged occluders compared to a rectangle-based masking. Note that the shape of the mask will further change when accounting for the land cover information. Therefore, both hard edges and more natural-looking patterns will be present in the final mask.

While these masks do not have the exact shape of change we would expect to see in reality, the randomness in them helps us ensure that the model is not learning the masking patterns. Additional, more natural masks could be generated, as we will discuss in Section 6 but we believe these must be coupled with a random mask-generating technique to prevent the model from learning spurious information.

With a probability of 75%, we are also applying simple augmentations individually to the original and modified images. The augmentations we are currently considering include colour horizontal and vertical flip, contrast, brightness, saturation, and hue adaptations.

Note that since the patches we sample cover an effective area of 32^2 m, the masks we generate are likely going to bias the model towards mid-sized changes.

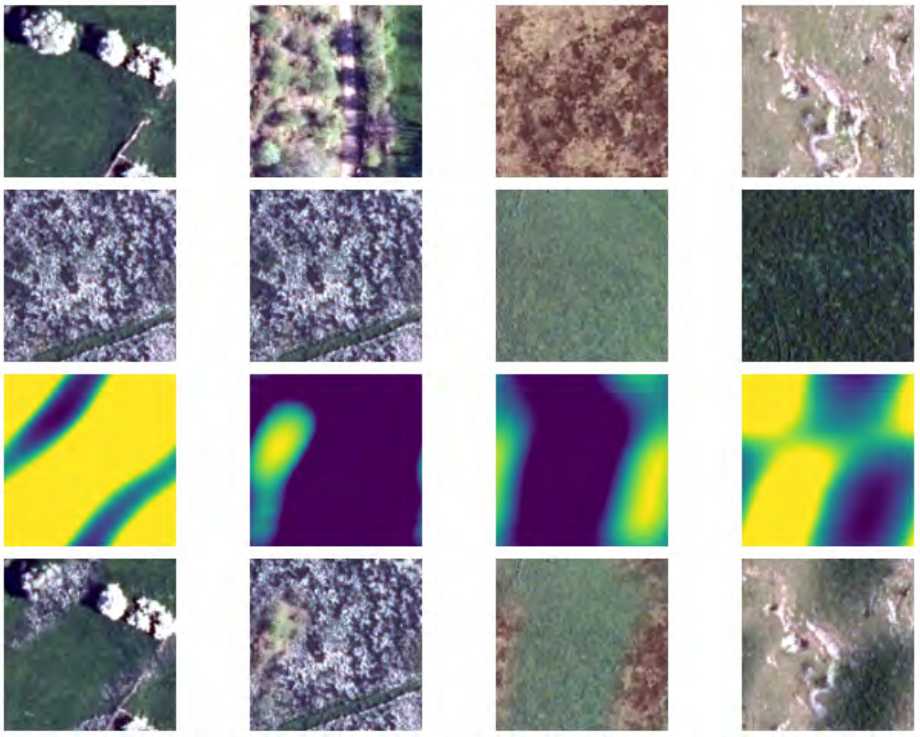


Figure 15: Examples of artificially generated samples. The first two rows contain the images to be mixed. The third row depicts masks sampled from Fourier space, while the last row shows the synthetic image obtained.

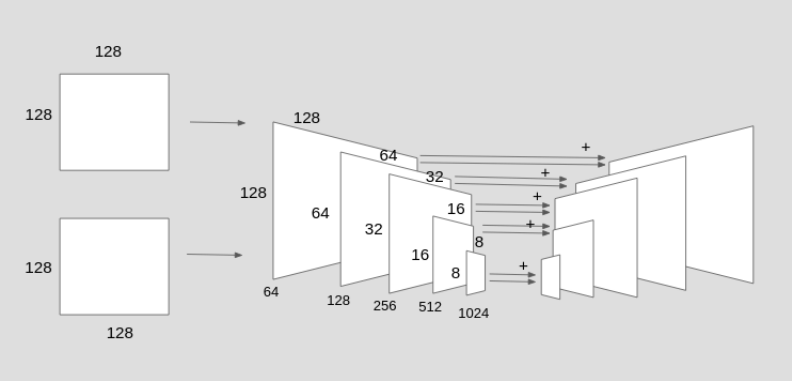


Figure 16: U-Net-like architecture used. Images are fed separately through the encoder and their extracted feature maps are combined at a decoder level.

5.7.1 Architecture

We used a U-Net[11]-like architecture and based our implementation on a publicly available repository. Compared to the usual segmentation use of a U-Net, we create an adaptation that passes the original and the modified image through the same encoder and then combines their features at the decoding phase. For more details on the architecture see Figure 16. We believe there are multiple possibilities for combining the features, which include adding, subtracting, or concatenating. Due to time constraints, we only explored subtracting the feature maps.

5.7.2 Experimental setting

Models were trained for 100 epochs, with a batch size of 32. We experimented with the AdamW, SGD with momentum, and AdaDelta optimisers and considered consine annealing, drop on plateau and multistep learning rate schedulers.

5.8 Analysis of preliminary results

Due to the intensive image modification, the initial model we trained was biased towards predicting that change has occurred even when no class change was present. To counter this, we applied the image mixing with a

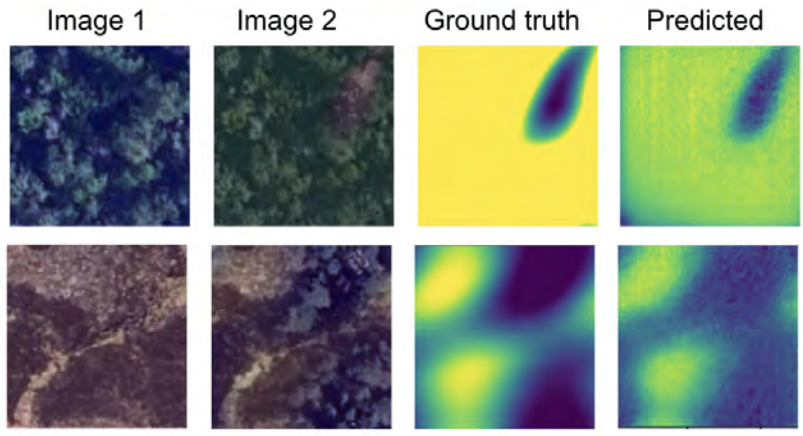


Figure 17: Model predictions on synthetic validation data. The model learns to correctly identify the class change.

probability of 50% so that the model would be trained on a more balanced data set. This has significantly improved training and allowed us to obtain a model with $\sim 92\%$ accuracy on the synthetic test data.

From the available data that had land cover information available, we do a 70-30 train-validation split. Figure 17 shows examples of results we obtain on the validation data, which visually confirm the high accuracy we report. The high accuracy we obtain indicates that the model does not simply learn to do a pixel-wise comparison of two images but instead it is able to identify whether the pixels that have changed belong to the same type of land cover or not. Therefore, we believe this provides strong evidence that the proposed approach is promising.

However, simply applying our model to the 2010 data requires more careful training. Firstly, since we needed to use the land cover information, the size of the training data was drastically reduced. This directly affects the model's invariance to the slight covariate shift we expect to see between the 2010 and 2020 data. For example, the model seemed to be sensitive to the shadows. To counter for this, we added an artificial shadow by sampling a Fourier mask with a 50% probability and applying gray patches on images. Note that unlike the mask used for mixing images, which is smooth (i.e. non-binary), for mimicking shadows we use a binary mask. We were unable to report full results on this

version due to time constraints but strongly believe it will further help the model learn invariance to spurious features.

Overall, we believe this approach is worth exploring in more detail and we are confident that with carefully designed improvements a robust model could be trained.

5.8.1 Future work

One possible way of reducing the covariate shift would be to also train with images from other years or regions. However, this must be carefully designed since without a rough land cover segmentation, a lot of incorrect data could be introduced. As we discussed earlier, mixing two images from the same class and setting the change map ground truth to the mixing mask would simply teach the model to predict pixel-wise change. An option to consider is including generating rough segmentation masks with the existing supervised approach. Although the supervised segmentation predictions might contain inaccuracies, we believe that this would make us less likely to create incorrect synthetic data. Nonetheless, we believe a relatively small percentage of data where the land cover was not annotated by experts should be used and additional methods for mitigating the shift should be considered.

There is a great scope for future work in terms of architecture definition. An immediate alternative for example is to adapt a U-Net with ResNet [8] backbone for our setting. Similarly, due to time constraints we did not experiment extensively with the optimisation side of training. We believe significantly better results can be obtained by doing so.

Lastly, when making predictions, a small artefact seems to consistently appear in the corners of the change mask. We believe this is due to the padding in the network and can be straightforwardly removed in future.

5.9 Auto encoders for change detection

An autoencoder is a type of neural network frequently employed for representing data in a compressed form [4]. Its architecture comprises an encoder and a decoder, as seen in Figure 18. The encoder compresses

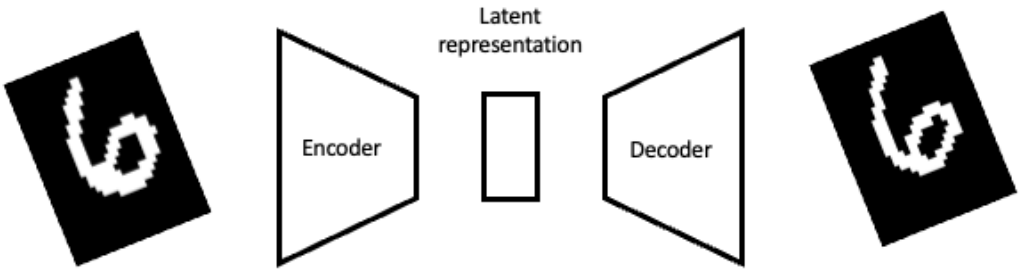


Figure 18: Autoencoder architecture schematic.

the data to distil only meaningful information, mapping each input data sample to a lower-dimensional representation in latent space. This latent space representation forms the compressed version of pertinent feature vectors. Subsequently, the decoder reconstructs the original input from this latent space representation. The primary objective of an autoencoder is to learn an efficient representation of the input data by minimizing the reconstruction error between the input and the output. The auto-encoder reconstructs only the most significant features by compressing data into the latent space. Consequently, this facilitates the representation of complex structures in the data as simplified form. Autoencoders have been applied to various problems such as dimensionality reduction, denoising, and anomaly detection ...They usually work well in unsupervised learning problems [10].

Application in Satellite Image Change Detection

Change detection in satellite images is a vital operation in the realm of remote sensing, necessitating the recognition and analysis of alterations within distinct snapshots of the same geographical area (Peak Districts) captured at different time points. Autoencoders have found wide-ranging applications in various computer vision tasks related to satellite imagery, including land crop analysis and weather prediction [3],[2]. These neural network models are powerful tools for discerning underlying patterns in

complex and feature-rich land cover satellite images.

In this study, we utilized an autoencoder to identify changes in the Peak Districts by processing two images from two separate time points: 2010 and 2020. The reconstructed image from the autoencoder acts as a conduit for evaluation. In this pipeline, images from different epochs are input, and a change map—consisting of the differences in learned features—is generated. This map provides a graphical representation of the regions where significant changes have occurred over the course of time.

The workflow for using our auto-encoder in change detection for the images from the peak district.

1. **Data Preprocessing:** The original images, which were of dimensions 8000 x 8000, were resized to a smaller resolution of 128 x 128. The cropping was the sole preprocessing step implemented, as the original dataset had already been subjected to thorough cleaning.
2. **Training Architecture:** We have undertaken the task of training an autoencoder using the available satellite imagery of the Peak District. This autoencoder learns to transform the input images into a compressed representation in the latent space and then reconstruct the original images from this condensed form. In our experiments, we have tested the model under diverse configurations for a more comprehensive understanding of its performance. In one instance, the architecture includes three convolutional layers and a single fully connected layer. In a separate variation, we employed four convolutional layers in both the encoder and the decoder parts of the model without any fully connected layers. However, there were no significant variations in the results.
3. **Results:** Once the auto-encoder is trained, it can encode and reconstruct new pairs of satellite images captured at different time points. The differences or changes between the reconstructed images can be identified by comparing the outputs from the autoencoder.

Autoencoders confer numerous benefits when employed in satellite image change detection. Their ability to autonomously decipher intricate

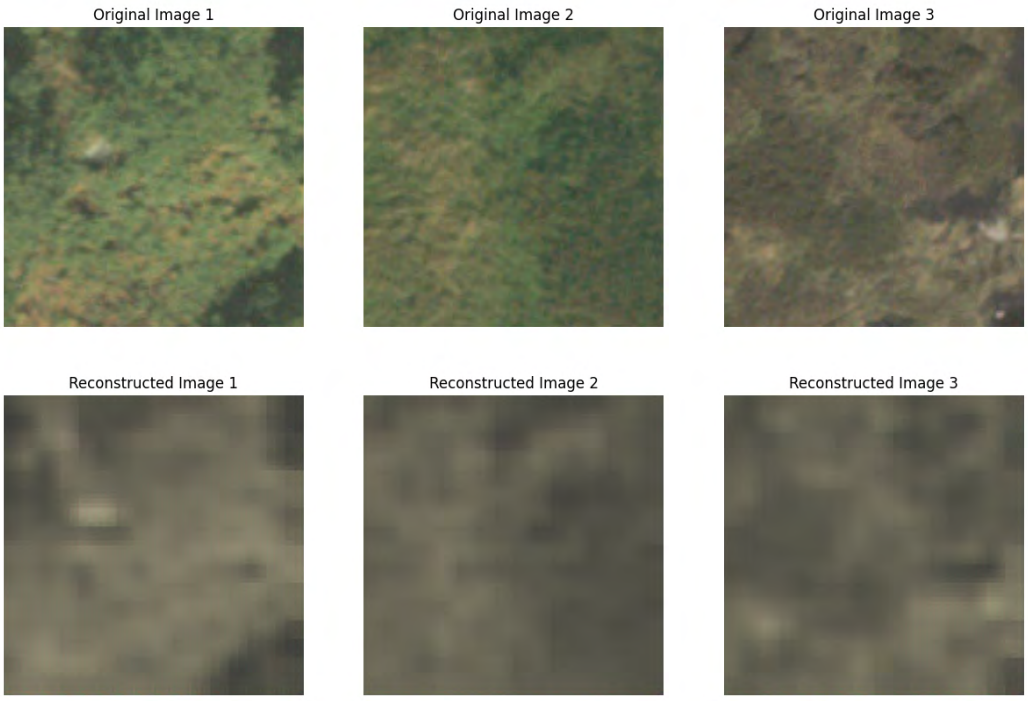


Figure 19: Autoencoder without fully connected layers

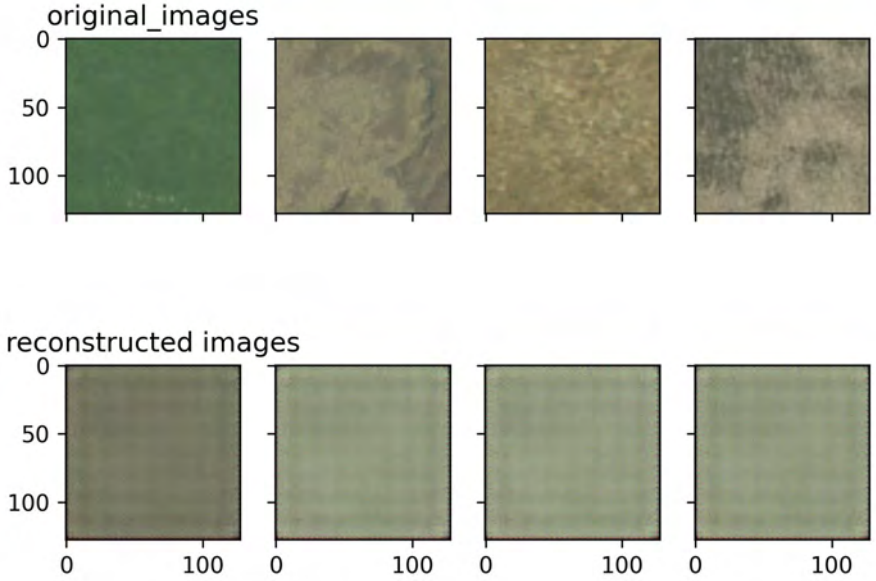


Figure 20: Comparison of original and reconstructed images for an Autoencoder with a wide kernel size (8) in the initial convolutional layer, connected to two consecutive layers (kernel size = 3) and a fully connected layer and output to a latent space with 20 dimensions

spatial and spectral patterns within images allows them to detect subtle modifications often invisible to the naked eye. Moreover, as learned by the autoencoder, the concise and insightful representation of images in the latent space contributes to effective and efficient change detection and subsequent analysis.

However, the current stage of our results didn't meet our expectations. The comparisons highlighted below, conducted on networks with three convolutional layers, didn't yield the anticipated output. Yet, we are optimistic that fine-tuning the hyperparameters could potentially enhance the performance of these models.

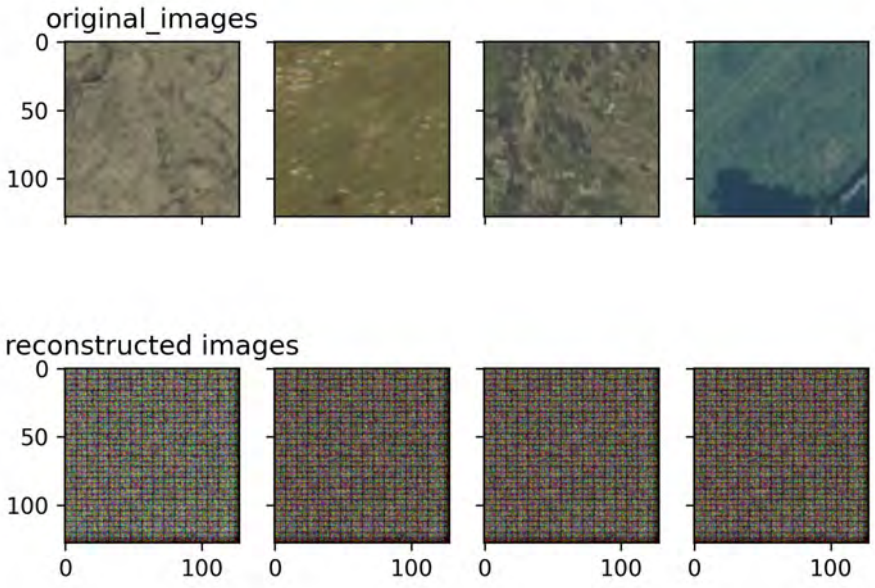


Figure 21: Comparison of original and reconstructed images for an Autoencoder with narrow convolutions (kernel size = 3) in three consecutive layers, before a fully connected layer and output to a latent space with 256 dimensions

5.9.1 Variational Autoencoder(VAE)

is a type of generative model that, unlike traditional autoencoders, maps each data point in the input to a distribution in the latent space rather than a single point. This facilitates the generation of new data samples but introduces an element of uncertainty into the encoding-decoding process. We explored using a VAE for this task; however, due to time constraints, we only worked on a surrogate MNIST dataset consisting of binary image data. To use it on our data, the VAE must be adapted to be used for predicting RGB data, modifying the number of channels and the reconstruction distribution to allow for the appropriate data format. This could be tried as a possible extension of the project.

Various variations of the auto-encoder were attempted. These included

5.9.2 Hyper parameters

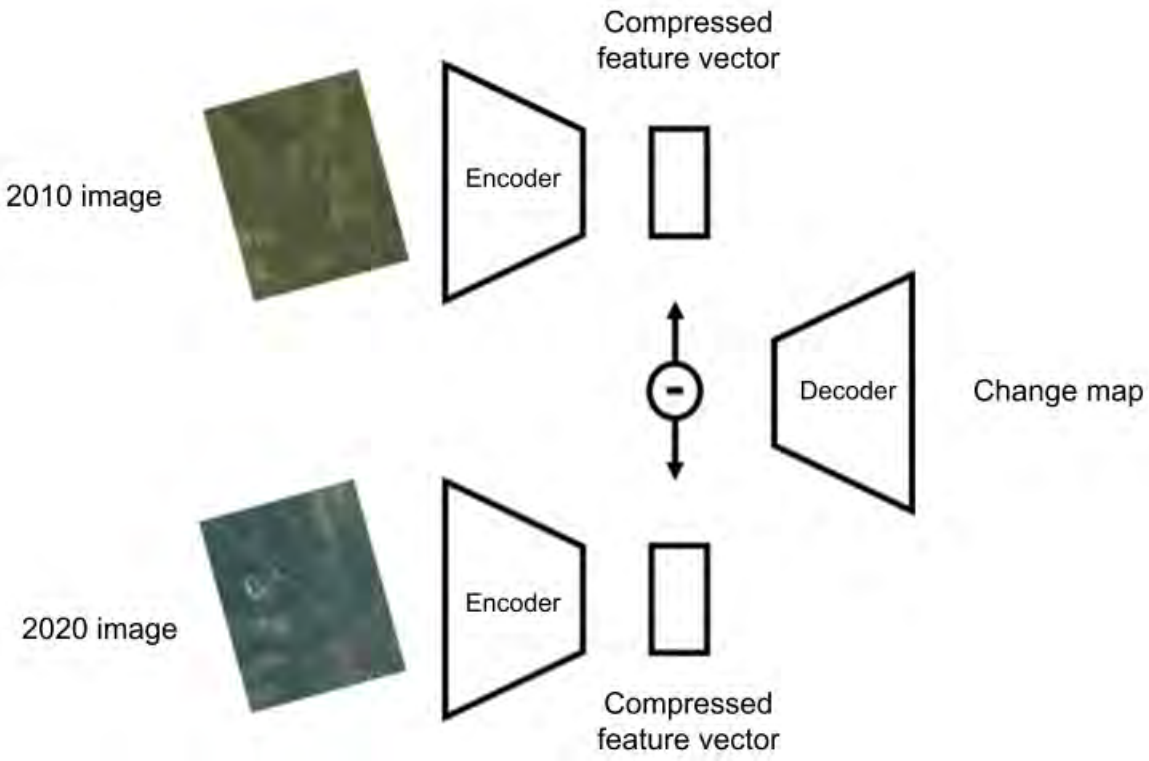
Table 3: Autoencoder Training Parameters

Parameter	**Value**
Learning Rate	0.001
Batch Size	128
Number of Epochs	50
Optimizer	Adam
Loss Function	Mean Squared Error
Encoder Layers	3
Decoder Layers	3
Activation Function	ReLU

5.9.3 Image differencing in latent space

The autoencoder was used to produce change maps with feature maps incorporating information from the entire dataset, unlike statistical and KPCA-MNet methods. The autoencoder was trained on examples chosen from the datasat of both 2010 and 2020 aerial images, to capture the

Figure 22: Image differencing in latent space. ©Bluesky International Limited and Getmapping Plc [2011, 2020]



5.9.4 Future Works

We have explored a number of auto-encoder architectures, which show some promising results, other avenues that exploring would be fine-tuning also future research direction might attempt to explore other state of art architectures such as U-net and Vision Transformers.

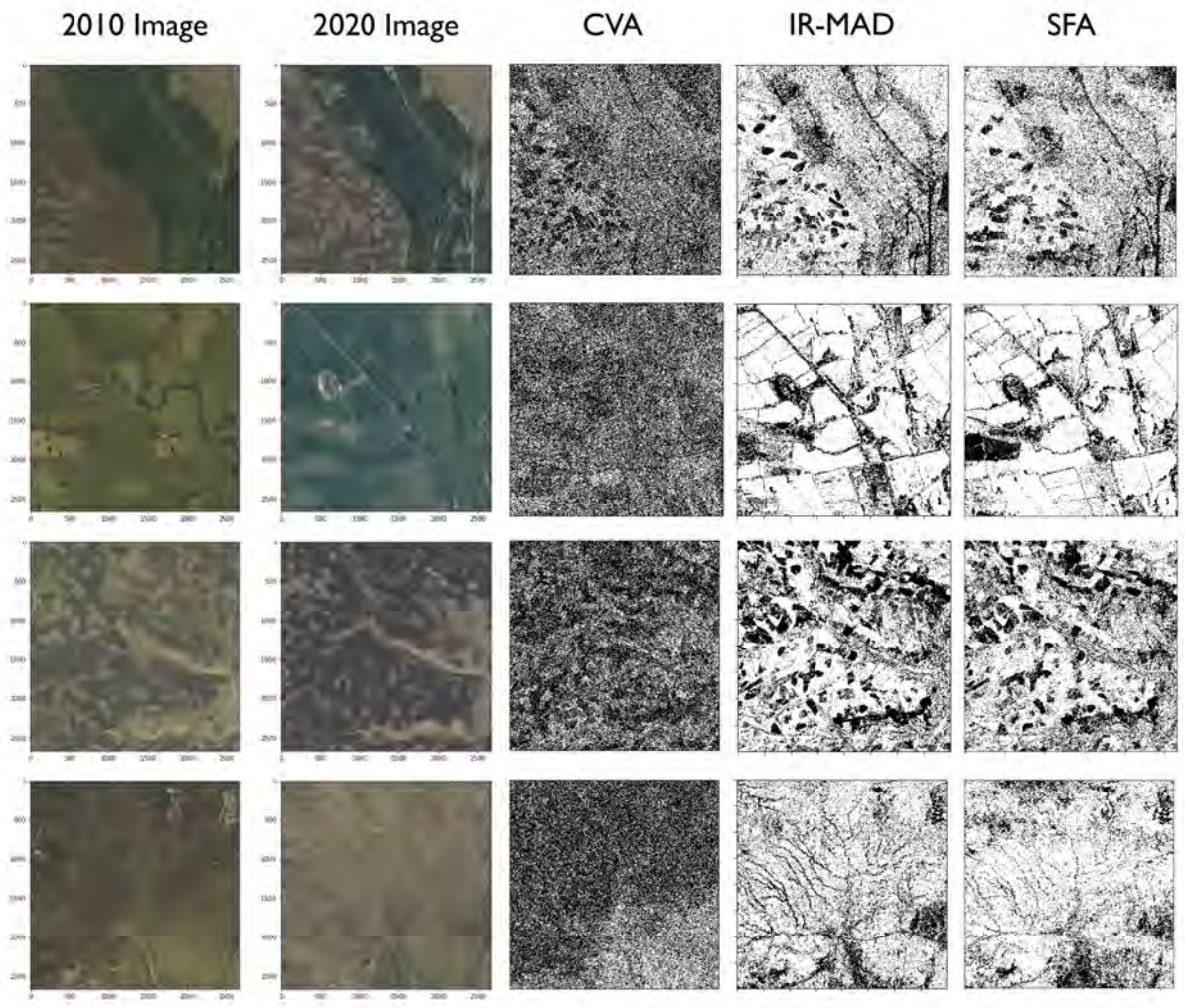


Figure 23: Change map outputs for RGB images: CVA, IR-MAD and SFA,
© Bluesky International Limited and Getmapping Plc

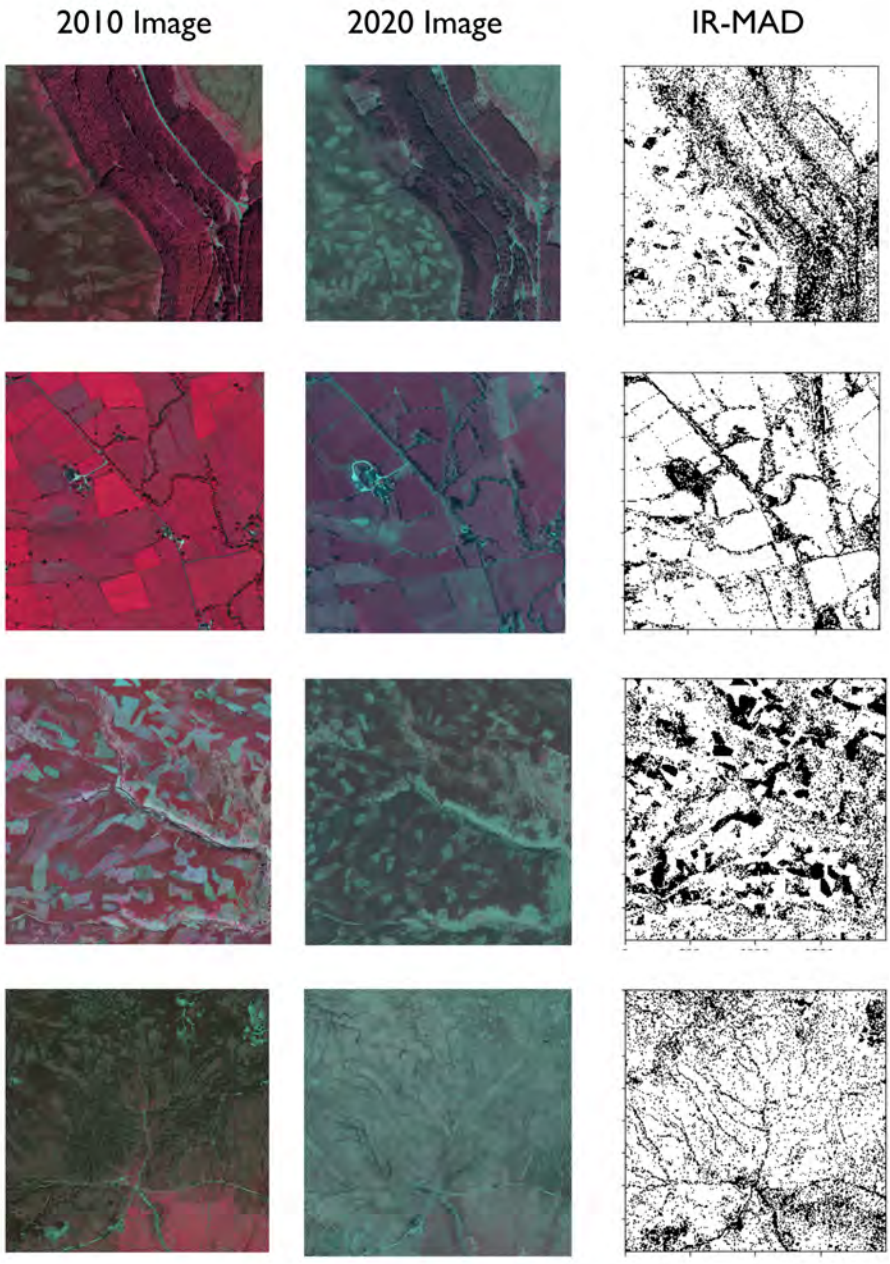


Figure 24: Change map outputs for IR images: IR-MAD, © Bluesky International Limited and Getmapping Plc

6 Discussion and Future Work

6.1 Discussions

To employ some of the more cutting edge unsupervised SNNs that have already applied to aerial photography would require correspondence with authors of the relevant papers to get code working or access pre-trained models. Given the current rate of development, it is likely that similar approaches will continue to influence the field and it may be fruitful to pursue in house development of such methods.

Neural network training (Autoencoders specifically) could be accelerated by optimising image loading as current code must load and then crop images from 8000 to 128 pixels. There are potential time savings to be made by caching a small number of images and using several patches within each for every training batch, rather than taking the time to load and then discard the majority of image data.

6.2 Future Work

Future work should focus on some of the limitations that were identified in this work. Users should pay close attention to any comparisons made when the acqusition months vary significantly between the two years. This is because particular species such as bracken changes its colours significantly between summer and winter months.

In addition, these current techniques do not take into consideration the different times of day that the acquisitions were taken. This means that shadows from objects in the image can potentially introduce artefacts of changes in the land usage.

6.3 Recommendations

One of the main challenges of change detection is to disambiguate what pixel change is relevant land cover change and what is noise (due to, e.g., seasonality, sensor normalisation and shadows). The statistical methods (CVA, IR-MAD, SFA) seem fundamentally limited in overcoming this noise problem, while deep learning based methods (Resnet feature extraction,

auto-encoders, possibly siamese networks) are in principle capable to overcome this issue.

Performance of deep learning based methods could be further improved by using encoders pretrained on the same or similar data, instead of using encoders pretrained on generic image data sets.

Synthesising change data (by mixing images) has the potential to provide the data required to train some of these deep learning methods.

7 Team members

Antonia Marcu is a PhD student at University of Southampton working on creating a data-centric theory of generalisation. Her research interests include robustness analysis, representation learning and data modification.

Mayank Sharma is an Associate National Project Officer in Data Science at UNESCO MGIEP.

Alex Milne is a 3rd year PhD student studying at Swansea University. His current work is with 2D sensor data in the steel manufacturing process, regressing surface properties of steel during/after the Temper Rolling process.

Jennifer Stiens ('Jen') is a 3rd year LIDo/Bloomsbury fellowship PhD student at Birkbeck University of London studying the non-coding genome of the Mycobacterium tuberculosis complex and how it may affect host specificity, using mostly computational methods. Her contributions were mainly in the areas of data investigation, report writing and asking lots of questions!

Alicja Polanska is a PhD student in astroinformatics at University College London's Data Intensive Science CDT. Her research focuses on machine learning applications in Astrophysics.

Stephen Town is a senior research associate in the Faculty of Brain Sciences at University College London.

Nurul Abedin is a research student in the Aeronautics and Aerospace research centre at City, University of london. His current research aims

to develop a data-driven crystal plasticity constitutive model to accelerate crystal plasticity simulation.

Ben O'Driscoll is a Data Visualisation Developer based at Plymouth Marine Laboratory (PML)

Wenlan Zhang is a PhD student in advanced spatial analysis at the Centre for Advanced Spatial Analysis, UCL, working on mitigating digital injustice with data fusion, with a case study in flood risk assessment in Nairobi, Kenya.

Sokipriala Jonah is a PhD researcher at the Centre for computational science and mathematical modelling Coventry, his research focuses on the improvement of next-generation wireless networks using machine learning.

Miguel Espinosa is a PhD researcher at the University of Edinburgh SENSE CDT. His research lies in the intersection of self-supervised computer vision and remote sensing.

References

[1] Yasir Afaq and Ankush Manocha. “Analysis on change detection techniques for remote sensing applications: A review”. In: *Ecological Informatics* 63 (2021), p. 101310.

[2] Heewoong Ahn et al. “Searching similar weather maps using convolutional autoencoder and satellite images”. In: *ICT Express* 9.1 (2023), pp. 69–75.

[3] Giuseppina Andresini et al. “Leveraging autoencoders in change vector analysis of optical satellite images”. In: *Journal of Intelligent Information Systems* (2022), pp. 1–20.

[4] Dor Bank, Noam Koenigstein, and Raja Giryes. “Autoencoders”. In: *arXiv preprint arXiv:2003.05991* (2020).

[5] Hongruixuan Chen et al. “Change detection in multi-temporal vhr images based on deep siamese multi-scale convolutional networks”. In: *arXiv preprint arXiv:1906.11479* (2019).

[6] Hongruixuan Chen et al. “Change detection in multisource VHR images via deep siamese convolutional multiple-layers recurrent neural network”. In: *IEEE Transactions on Geoscience and Remote Sensing* 58.4 (2019), pp. 2848–2864.

[7] Ethan Harris et al. “Understanding and Enhancing Mixed Sample Data Augmentation”. In: *arXiv preprint arXiv:2002.12047*. 2020.

[8] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[9] Simon Kornblith et al. “Similarity of neural network representations revisited”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 3519–3529.

[10] Umberto Michelucci. “An introduction to autoencoders”. In: *arXiv preprint arXiv:2201.03898* (2022).

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, pp. 234–241.

[12] Attaullah Sahito, Eibe Frank, and Bernhard Pfahringer. “Semi-supervised learning using Siamese networks”. In: *AI 2019: Advances in Artificial Intelligence: 32nd Australasian Joint Conference, Adelaide, SA, Australia, December 2–5, 2019, Proceedings 32*. Springer. 2019, pp. 586–597.

[13] Qiuhong Shen et al. “Unsupervised learning of accurate Siamese tracking”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 8101–8110.

[14] Xian Tao et al. “Unsupervised anomaly detection for surface defects with dual-siamese network”. In: *IEEE Transactions on Industrial Informatics* 18.11 (2022), pp. 7707–7717.

[15] Chen Wu et al. “Unsupervised change detection in multitemporal VHR images based on deep kernel PCA convolutional mapping network”. In: *IEEE Transactions on Cybernetics* 52.11 (2021), pp. 12084–12098.

[16] Sangdoo Yun et al. “CutMix: Regularization strategy to train strong classifiers with localizable features”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 6023–6032.

[17] Hongyi Zhang et al. “mixup: Beyond Empirical Risk Minimization”. In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=r1Ddp1-Rb.

[18] Wenwei Zhang et al. “Dense Siamese Network for Dense Unsupervised Learning”. In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX*. Springer. 2022, pp. 464–480.

The
Alan Turing
Institute

turing.ac.uk
@turinginst