

Article

Using Machine Learning to Map Western Australian Landscapes

Thomas Albrecht, Ignacio González-Álvarez* and Jens Klump

CSIRO Mineral Resources, Australian Resources Research Centre, 26 Dick Perry Ave., Kensington, Perth, WA 6151, Australia

* Correspondence: ignacio.gonzalez-alvarez@csiro.au;

Abstract: Landscapes evolve due to climatic conditions, tectonic activity, geological features, biological activity, and sedimentary dynamics. These processes link geological processes at depth to surface features. Consequently, the study of landscapes can reveal essential information about the geochemical footprint of ore deposits at depth. Advances in satellite imaging and computing power have enabled the creation of large geospatial datasets, the sheer size of which necessitates automated processing. This study describes a methodology to enable the automated mapping of landscape pattern domains using machine learning (ML) algorithms. From a freely available Digital Elevation Model, derived data, and sample landclass boundaries provided by domain experts, our algorithm produces a dense map of the model region in Western Australia. Both random forest and support vector machine classification achieve about 98% classification accuracy with reasonable runtime of 48 minutes on a single core. We discuss computational resources and study the effect of grid resolution. Larger tiles result in a more contiguous map, while smaller tiles result in a more detailed, and at some point, noisy map. Diversity and distribution of landscapes mapped in this study support previous results. In addition, our results are consistent with the geological trends and main basement features in the region.

Keywords: supervised machine learning, automated landscape mapping, digital elevation model

1. Introduction

The landscapes of Western Australia are characterised by low topographic relief and a deeply weathered blanket that covers most of the State. This landscape context is challenging for mineral exploration due to the lack of direct fresh rock exposure at surface. This is common to large regions in China, West Africa, Brazil, South America and most of Australia [1, and references therein]. However, domain experts have identified distinct landscape variability that changes imperceptibly at small-scale. Landscape evolution may record geochemical and stratigraphic information that link the basement and the surface [e. g. 2–7]. The integration of landscape evolution and surface geochemistry in deeply and intensely weathered regions is a fundamental tool in mineral exploration used around the world [e. g. 8–10, and references therein].

Landscapes are defined by their surface features, such as landforms, vegetation, slopes, sedimentary systems, soils, etc. [11]. The variability of these features allows landscapes to be differentiated one from another, which facilitates classification. From a geological perspective, landscapes are fundamentally linked to: (i) climatic conditions, which control, e.g. water availability and weathering processes; (ii) tectonic activity, a main driver of the evolution of surface elevation evolution which also has a significant impact on sedimentary dynamics and erosional processes; (iii) geological features, such as structures and lithologies, which control the architecture and variability of the land at surface and at depth [2–6]; and (iv) tectonic activity that controls the flow of metal-bearing fluids that may lead to the formation of mineral deposits [12]. Consequently, studying landscapes can link surface features to geological processes. In this work we asked the question whether these subtle differences between landscape types in low-relief and intensely weathered regions can

be quantified and whether the definition of basic landscape classes can help identify deep-seated geological anomalies that may have acted as conduits and hosts in the formation of mineral deposits.

2. Overview of technologies with remote sensing data for landscape mapping

For small scales of approximately 10 km, a wide variety of quantitative methods have been developed in geomorphology [13–17, and references therein]. Others have applied quantitative methods successfully at regional and continental scale [e.g. 18–20]. The difficult lies both in identifying features that can differentiate landscapes into different domains, and in using those features to determine the geographic extent of the domains. This is exacerbated in a setting where landscapes are difficult for the untrained eye to distinguish, as is the case of the deeply weathered landscapes of Western Australia.

One of the most challenging aspects of landscape study is cataloguing their differences at regional or even continental scale [e.g. 21]. The difficult lies not in identifying features that can differentiate landscapes into different domains, but in using those features to determine the geographic extension of the domains. Field observations have been crucial to understand landscape diversity at regional scale [7]. However, field observations are practically limited to the immediate vicinity of roads and hence inherently linear, and any attempt at extrapolating to regional scale introduces significant uncertainty. Such extrapolation suffers from the complexity and variability of landforms, and the difficulty in defining quantitative criteria that discriminate diverse landscape types.

Advances in satellite imaging and computing power have enabled the creation of huge yet accessible geospatial datasets. These datasets can assist in characterising landscape features and their distribution at regional scales [e.g. 22–24], which directly impacts many disciplines, including mineral exploration, geomorphology, environmental studies and geoarchaeology, amongst others [e.g. 8–10,25]. Moreover, combined with the push towards ML in recent years, such datasets have enabled a data-driven approach to automatic landscape classification, without explicitly deriving a specific metric first. Presented with a set of labeled data, a ML classifier learns how to distinguish between classes and can then predict labels for unseen data. Support Vector Machine and Random Forest classifier typically produce the most accurate results. State-of-the-art in the field of Computer Vision, Deep Convolutional Neural Networks (DCNN) are becoming more common in geology as well. DCNN implicitly aggregate, require little to no feature engineering [26], and classification accuracy often exceeds 90%. However, training is computationally very expensive.

Classification for mapping purposes is naturally based on spatial input data, which lends itself to either a pixel or an object approach [27]. Conceptually simple, the pixel approach directly feeds pixel values into ML. However, since a pixel is a largely arbitrary region from geographical point of view, the pixel approach is typically less accurate as it disregards spatial context.

By contrast, an object is a geographically meaningful region and therefore respects natural borders (e.g. roads, for urban mapping). Pixel values from said region are aggregated before being fed into the ML classifier, preferably in a way that preserves some spatial context. This aggregation may improve accuracy, but is more complex to implement.

Most studies of land cover/land use classification or urban mapping use the pixel approach, and typically achieve accuracies around 80–90%, sometimes 95%. Lidberg *et al.* [28] show that ML outperforms previously used single thresholds when creating high-resolution wet area maps. They report an overall accuracy of 84% for RF and Artificial Neural Networks (ANN), followed by 82% for SVM, as compared to 79% for single-threshold classification. Maxwell *et al.* [29] compare the performance of 6 ML classifiers for an agricultural and an urban land cover data set, and report 89.1% accuracy for SVM and 87.1% for RF. Feature selection (pick only the most important features) increases accuracy to 94.4% for SVM, while decreasing that for RF to 87.8%. They also “recommend that, if possible, the analyst should experiment with multiple classifiers to determine which provides the optimal classification for a specific classification task.” Similarly, Abdi [26] compares SVM, RF, extreme gradient boosting, and DCNN for land cover and land use classification in a boreal landscape. SVM

performs best in his study, however, all approaches yield very similar accuracies in the range of 76% - 73%. For urban mapping, Zhang *et al.* [27] combine remote sensing data and social sensing data, and report 78% accuracy using a random forest classifier.

An interesting approach that fits the landscape mapping category but doesn't make use of canonical ML classifiers is that of Jasiewicz et. al [19,30]. They first compute a signature for a local region (e.g. a 16×16 px block) called a scene. Then they compute the distance between scene signatures. Their library provides several signature functions and distance metrics. "Being able to quantify the overall similarity between two landscapes using a single number is a key element" of their work, which allows them to use averaging for classification resulting in excellent run-time performance. Further, how their algorithm decides could also be easily reproduced and validated by a human. However, reducing landscape complexity to a single dimension inevitably leads to some information loss. In connection with various averaging steps employed in their methodology, this may or may not be critical. They don't report any accuracy metric.

The literature suggests that landscape mapping algorithms based on classic ML are often less accurate than those using DCNN, perhaps because most studies follow the pixel approach, i.e., don't aggregate pixel values prior to classification. Our goal was to implement an algorithm that (1) performs comparable to ANN/DNN (i.e. accuracy $\geq 90\%$), but is based on classic ML methods for which training is cheap; (2) is open source, written in a modern language, therefore easily extendable; and (3) can potentially be run in parallel, enabling processing on a continental scale.

We apply our algorithm to a region of approximately $1\,000 \times 1\,000 \text{ km}^2$ in the south of Western Australia, see figure 1. Inputs are a freely available¹ Digital Elevation Model (DEM) from Geoscience Australia [31] along with derived data, and sample landclass boundaries provided by domain experts. Our algorithm then produces a dense map of the model region.

In the following section 3, we review a general ML workflow and detail our methodology. In section 4 we study how tile size affects classification accuracy, and demonstrate a number of extensions that may improve accuracy or provide additional functionality. Discussion and conclusion are presented in section 5.

¹ licensed under the terms of the Creative Commons Attribution 3.0 Australia license

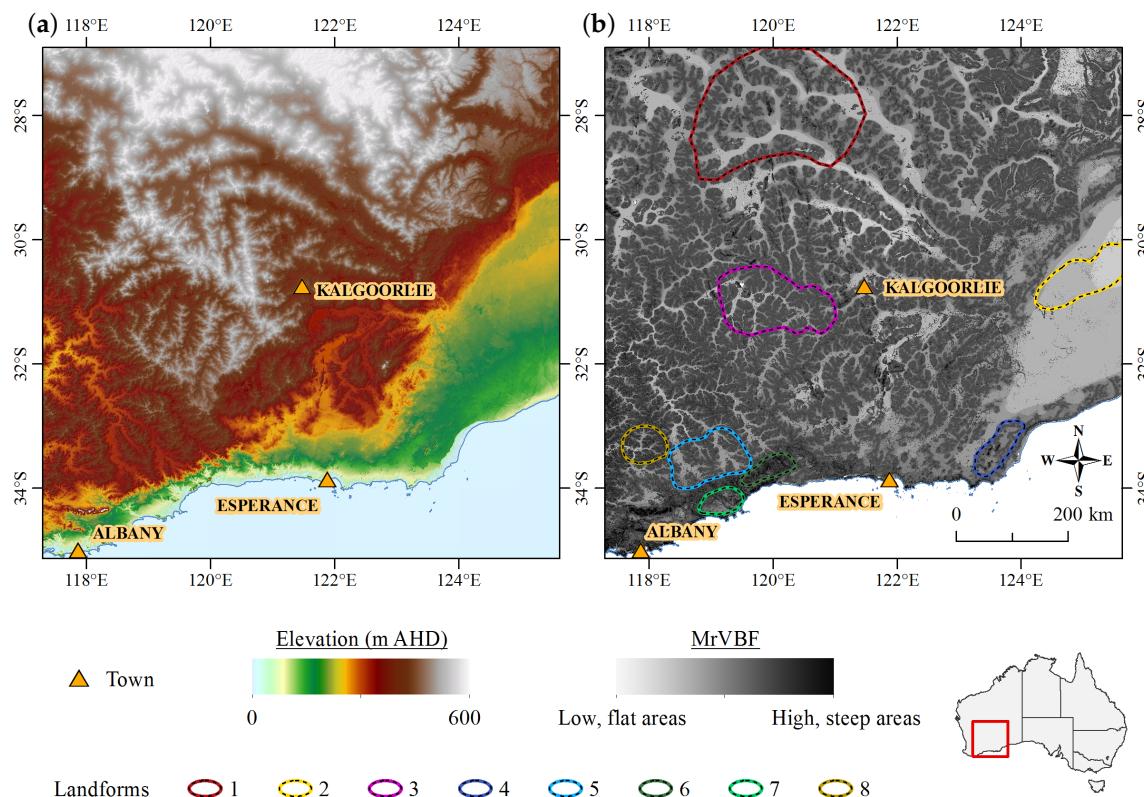


Figure 1. (a) DEM [31] of the study area. (b) Multi-resolution valley bottom flatness map [MrVBF, 32], a sub-product of the DEM, that displays low slope gradients in lighter shade and high slope gradients in darker shade. Colour indicates the eight flatness surface patterns used in this study.

3. Methods

3.1. Machine Learning

Traditionally, for a computer to solve a problem, we *explicitly program* a series of instructions, largely of type “if this, do that”, tailored to said problem. In ML, instead of explicit programming we *train a model*. One method of training is to provide known input data along with known good decisions. After successful training, the model is able take good decisions for new, unseen data. Known as supervised learning, we will use this method in our paper.

The main advantage over explicit programming is that ML models can learn very complex relationships between data and decisions (provided enough good training data is available), which otherwise would require a human to implement similarly complex —hence costly and error-prone—algorithms. Even worse, said relationships may not even be known, rendering explicit programming impossible.

A general ML workflow includes three main steps: (i) *preparing* the source data, i.e. selecting data that we think is relevant to the problem at hand, and transforming it into a format suited for (ii) *training* a model, followed by (iii) *predicting* new, unseen data [33]. Machine learning methods can be split into two approaches: supervised and unsupervised learning. Supervised learning requires labeled data on which a model can be trained; such labels are usually provided by a human domain expert. Unsupervised learning requires no labels; one goal could be to detect clusters—i.e. groups with similar features—within the data. We limit this study to supervised learning, but note that an extension to unsupervised learning is straightforward.

Our task is classification: predicting a label for a sample, given a set of features. As labels we chose arbitrary integers that uniquely identify the eight landforms indicated in figure 1. We have

tested a number of classification methods, namely (i) Support Vector Machine classifier with linear and radial basis function kernels, denoted SVC-LIN and SVC-RBF [34,35]; (ii) Decision Trees [DT, 36]; and (iii) Random Forests, consisting of 100 or 1000 Decision Trees [RF-100 and RF-1000, 37], all accessed via their python wrappers in scikit-learn [38]. Implemented in Python 3.7, our code is available at <https://gitlab.com/fgradi/lpr/>.

3.2. Data preparation: assembling the feature matrix

Many fundamental ML algorithms are designed to operate on a so-called feature matrix. Each row in this matrix corresponds to a sample of the quantity we want to predict. In our case, a sample is the smallest spatial unit which gets assigned a landclass label. We call such units “tiles” (cf. figure 2), and chose square tiles with side lengths ranging from 51 to 1001 px (4.4 to 86 km). Columns represent features: attributes of a tile which we think are relevant for its classification, e.g. slope or flatness. Assembling the feature matrix from our source data is the major and first of the three steps of our ML approach, which we will now describe in detail.

Since the task is 2-D pattern recognition and classification, it is crucial that (some) spatial context is preserved during this step. In initial exploratory analysis, we applied principal component analysis to the feature vectors for each pixel, with each pixel treated as an independent observation, and with no consideration of the spatial correlation between nearby pixels. When the pixel data were projected onto the principal components, we found that the component scores did not cluster according to the landscape classes, and the resulting map (not shown) was inaccurate and noisy. Also this approach produces a very large feature matrix which increases memory requirements.

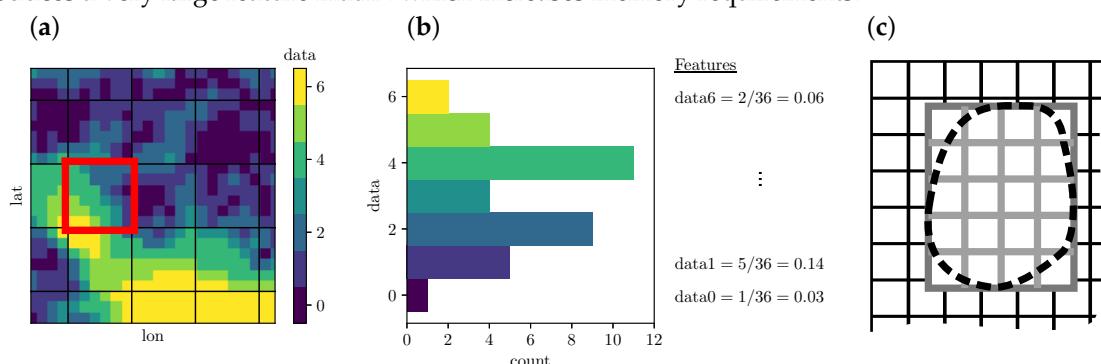


Figure 2. Spatial feature aggregation. (a) We decompose the source data into tiles of size $s \times s$. (b) Within each tile we compute the fraction of each area that is equal to a possible value. In other words, we compute per-tile normalised histograms of source data. In this example, we use $s = 6$ and a generic field of ‘data’ ranging 0 to 6, sorted into seven bins, which results in seven features ‘data0’ ... ‘data6’. (c) The bounding box of a landclass training region determines the training grid (gray), which generally differs from the grid (black) used for predicting the final map. This helps to identify overfitted models. See text for details.

A more viable approach is spatial aggregation of our source data, but in a way that preserves variation of features typical for a particular landscape class. Simple averaging would not work as it smooths out those variations, leaving little to distinguish between classes. Instead, we decompose the data into tiles as shown in figure 2 and compute per-tile normalised histograms of source data. Tiles may overlap, and tile size s and stride r can be specified. Treatment then depends on the type of our numerical input, which is either nominal, ordinal, or integer/real.² For nominals, within each tile we calculate the fraction of each area that is equal to a possible value. For ordinals, treat as nominals

² Integer and real data have both ordering ($1 < 2 < 4$) and distance ($2 - 1 < 4 - 2$). Nominals have neither, e.g. “sunny”, “overcast”, “rainy”. Ordinals have an ordering, but no notion of distance: “cold” < “warm” < “hot”.

dataset	source
data type, min...max, features	
Geosciences Australia 3" SRTM DEM v01 integer, -71...2222, skewness, kurtosis, inter quartile range (IQR)	[31]
Multi-resolution Valley Bottom Flatness (VBF) ordinal, 0...9, normalised histogram (NH), 10 bins	[32]
Multi-resolution Ridge Top Flatness (RTF) ordinal, 0...9, NH, 10 bins	[39]
Topographic Wetness Index derived from 1" SRTM DEM-H (TWI) real, 2.897...22.795, mean	[40]
Slope derived from 1" SRTM DEM-S degrees, 0...76.94, skewness, kurtosis, IQR	[41]
Wind Exposition Index (WEI) real, 0.7713...1.314, mean, skewness, kurtosis, IQR	[42,43]
Slope Relief Classification derived from 1" SRTM DEM-S nominal, 6 slope codes (see table 2), NH, 6 bins	[44]

Table 1. Source data set, data type and range of a total of 37 features. Most features are generated by computing normalised histograms (NH), some are simple statistics such as mean, skewness and kurtosis.

LE	level
VG	very gently sloping
GE	gently sloping
MO	moderately sloping
ST	steep
VS	very steep

Table 2. Slope codes used by [44]

except using \leq instead of $=$ on each possible value. For integers and reals, discretise then treat as for ordinals.

Nominal or ordinal source data with n possible values results in n features. For integers and reals, the chosen discretisation determines the number of features generated. For example, the Slope Relief Classification source data consists of six slope classes of ordinal type, ranging from “level” to “very steep”. We generate six features slopeX: each value is the fraction of a tile that has the corresponding Slope Relief Classification.

Table 1 summarises all 37 features. We chose not use the DEM directly as we aim to classify shape, not elevation. We do however include basic statistics of the DEM as features, such as skewness, kurtosis, and inter quartile range. Normalising the data is useful as some ML algorithms are not scale invariant, i.e. perform poorly if some features range, e.g., from 0 to 1 while others range from 0 to 100. The code is designed to make adding additional features straightforward, for example one could easily include adjacency information as is done in [19].

The histogram approach has a number of advantages: (i) tile size directly controls the spatial scale of classification, (ii) model training is comparably cheap, potentially enabling continent-scale classification, (iii) classification is invariant under 90 degrees’ rotation, (iv) implementation is reasonably straightforward, and (v) how the algorithm decides can be easily validated by a human. Convolutional Neural Networks provide an alternative approach, and their applicability to landscape pattern recognition could be explored in the future.

3.3. Training, cross-validation, prediction

Once the feature matrix is assembled, we train a number of ML models. We evaluate their performance based on accuracy, defined as the number of correct predictions divided by the total number of predictions. In our example, accuracy is a suitable measure since the tile count per training class is roughly balanced, i.e., the data set is reasonably symmetric (see table 3). If this situation of symmetry did not hold up then other metrics might be appropriate, e.g. F1 score.

class label	1	2	3	4	5	6	7	8
number of training samples	831	583	3670	2036	871	1869	560	9940

Table 3. The number of training samples for our test region is roughly balanced.

Initially, we estimated accuracy by 10-fold cross-validation: Our labeled data is decomposed into ten equal-sized subsets. A model is trained using nine of these, and its performance is tested using the remaining subset. Repeating this process for different subsets yields ten accuracy estimates, the mean of which we refer to as cross-validation accuracy. Accuracy may be affected by the choice of hyperparameters, i.e. parameters which are unrelated to the training data, and which can be tuned during training. The hyperparameter space is typically small enough such that a simple grid search is feasible. Once we are satisfied with model performance, we train the final model using the entire training dataset, and then use it to predict a map.

As an alternative to cross-validation accuracy, one can evaluate accuracy post-training, i.e., based on said final prediction, for *all* tiles for which the ground-truth is known. This is usually considered bad practice, since accuracy would be based entirely on the same data that the model has been trained on.

In our case however, it was cross-validation accuracy that proved problematic. For tile sizes larger than 17 km, cross-validation accuracy would often approach 100%, whereas post-training accuracy for the same model was much lower, indicating over-fitting. This discrepancy is due to the fact that the grids we use for training are in general offset from the one on which we predict the final map, as shown in figure 2(c). For training, we extract a block of pixel data defined by the axis-aligned minimum bounding box of a landclass, then decompose that block into tiles (gray grid). The grid (black) for predicting the final map however is defined by the extent of the input DEM. While aligning both grids would be trivial, we chose not to, as this mismatch helps detecting models that are overly grid-dependent, i.e., overfitted. Post-training accuracy was almost always lower than cross-validation accuracy. We therefore chose to report only the conservative estimate of post-training accuracy, referred to as "accuracy" in the following.

3.4. Computational cost

In this section we report the computational cost in terms of aggregation, training and prediction CPU time, as well as memory requirements. All processing was carried out on an Intel® Core™ i7-8550U CPU notebook with 16 GB RAM.

Our test region extends from (117.2896, -26.8996) deg to (125.6246, -35.1346) deg, covering $794.2 \times 915.7 \text{ km}^2$. At 3 arc seconds resolution, raster images have a size of $10\,002 \times 9\,882 \text{ px}^2$. Decomposition into overlapping tiles using a *constant* stride size of 25 px produces $\approx 155\,000$ tiles of which $\approx 30\,000$ are invalid (ocean). A tile is considered valid if it contains at least 25% valid pixels. We emphasise that the total number of tiles is largely independent of tile size since we keep the stride size constant. This means that overlap increases with tile size, from 50% at the smallest tile size, to 75% at what we report below as the best tile size, to 97.5% at the largest tile size we studied. This decision was made in an attempt to keep the number of training samples constant, while still enabling us to study how aggregation length scale affects classification accuracy. We would expect that accuracy is low for small tiles as they fall short of the defining length scales. Increasing the tile size, without increasing

the overlap at the same time, would reduce the number of training samples, which typically degrades accuracy.

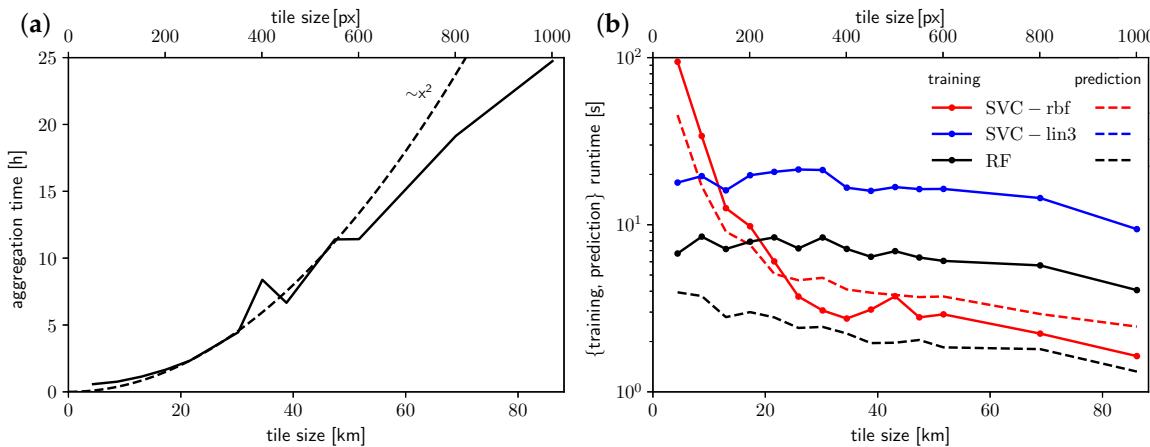


Figure 3. Effect of tile size on (a) aggregation, and (b) training and prediction runtime. Note that for clarity we omitted prediction runtime for SVC-LIN. It has a similar (lack of) dependency on tile size as RF-100, but is two orders of magnitude faster.

Figure 3 shows how aggregation, training and prediction runtimes scale with tile size. By far the largest cost, aggregation (figure 3a) takes between 1 and 25 h and scales approximately with tile size squared. This is expected as the total number of tiles is basically constant. We note that aggregation is an embarrassingly parallel problem, i.e. parallelisation should result in a linear speed-up. However, we did not attempt parallelisation of the process in this project.

Training (solid lines in figure 3b) is orders of magnitudes faster, typically taking seconds to minutes depending on the ML algorithm. As the size of the feature matrix is largely independent of tile size, so should be training runtime, which is indeed the case for RF-100 and SVC-LIN. For SVC-RBF however, training runtime varies significantly, from 1.6 s at $s = 86$ km to 95 s at $s = 4.4$ km, which may be due to the large overlap and hence, rather strongly correlated training samples at large tile size.

Finally, prediction runtime (dashed lines in figure 3b) generally follows the same trends: it is largely independent of tile-size for RF-100 and SVC-LIN, and strongly dependent on tile size for SVC-RBF. Prediction is about as fast as training for SVC-RBF, and about four times faster for RF-100. For SVC-LIN, prediction is 100 times faster than training, takes less than 0.05 s, and was omitted from figure 3(b) for clarity.

Memory usage varied between 1.8 and 2.2 GB. A larger workstation or cluster implementation should be sufficient to run the algorithm for the entire continent of Australia.

3.5. Hyperparameter tuning

Classification accuracy and runtime performance depend on the choice of hyperparameters: parameters specific to the ML algorithm, which are selected *before* training, and should be tuned for the problem at hand. Since training is very quick in our case, we conducted a simple grid search which produced an optimum $C = 10^6$ and $\gamma = 10^{-4}$ for SVC-RBF, and $C = 10^4$ for SVC-LIN.

For the Random Forest classifier, one has to specify the number of decision trees. We will report results for a Random Forest consisting of 100 decision trees (RF-100). We have also tested RF-1000, which increased training and prediction time by a factor of 10 (as expected), improved accuracy by merely 0.2%, and produced virtually identical maps.

4. Results

We visualise results by maps such as those in figure 4. Semi-transparent colours indicate the predicted landclass, dashed lines indicate training regions, and both are overlaid on top of the flatness map. A square (or circle) in the bottom right corner indicates the tile size. As explained above, tiles from the training set (i.e. those within dashed lines) may still be misclassified, which is the case if post-training accuracy is less than 100%.

4.1. Effect of tile size on accuracy

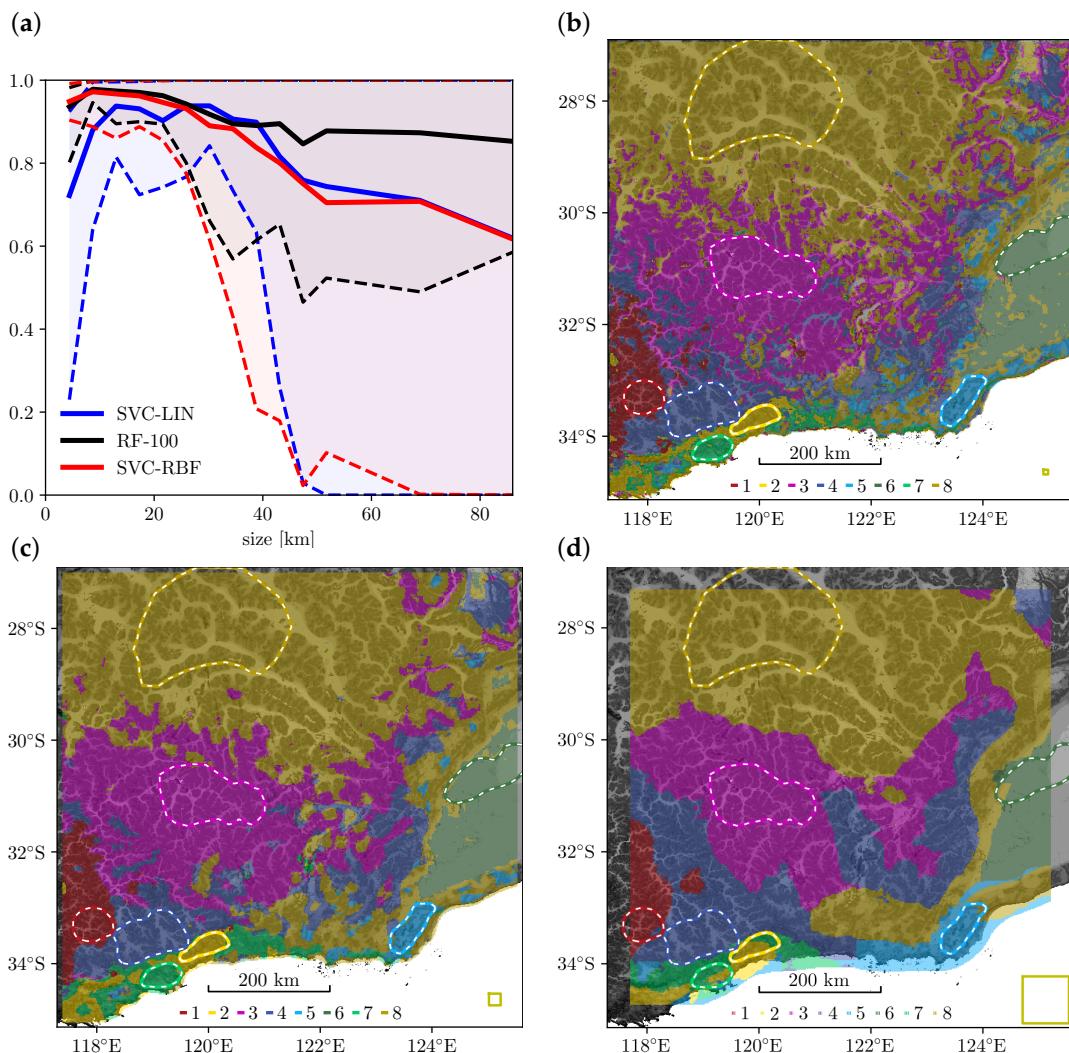


Figure 4. (a) Solid lines show the overall accuracy vs. block size, per ML algorithm. Shaded areas indicate the range of per-class accuracy. (b) The highest accuracy score of 97.8% is produced by RF-100 at $s = 8.7$ km, which is still a rather small tile size and results in a somewhat noisy map. The yellow square in the lower right corner represents a tile. (c) Increasing the tile size to $s = 21$ km reduces accuracy slightly to 96.3%, but produces a much cleaner map. (d) At the largest $s = 86$ km, a lack of training samples severely degrades accuracy for classes 2 and 7 whose training regions are small.

In this section, we will discuss the effect of tile size on the classification accuracy for different models. As expected, this effect is quite significant: “All spatial analyses are sensitive to the length, area, or volume over which a variable is distributed in space” [45]. In figure 4(a) we varied tile size from 4.4 to 86 km (51–1001 px), which produced an accuracy between 39.8 and 97.8%, shown by solid

lines, for each of our models. Shaded areas indicate the range of per-class accuracy, again per-model, which we will discuss further below.

On average, RF-100 (black line) produces the highest accuracy, followed by SVC-LIN (blue) and SVC-RBF (red). The best map, shown in panel (b), had an accuracy of 97.8% and was produced by RF-100 with a tile size of 8.7 km (101 px). This was the second smallest tile size we tested. The minimum per-class accuracy in this case was 94.6%, indicating excellent classification for every landclass.

The remaining algorithms achieve best accuracy at similarly small tile sizes. Accuracy peaks at $s = 13$ km for SVC-RBF, and $s = 17$ km for SVC-LIN. The resulting maps, shown in figure 5, are not fundamentally different from the overall best map, figure 4(b).

In general, smaller tiles increase the (nominal) map resolution and the number of available training samples, too little of which typically degrade accuracy. However, smaller tiles also increase time required for aggregation and training. More severely, smaller tiles might no longer capture the spatial scales which define a landscape for the domain expert. The result is a noisy map. Conversely, larger tiles reduce noise and generate smoother boundaries between classes: figure 4(c) shows a map with a tile size of $s = 21$ km, which still results in 96.3% accuracy. However, larger tiles also reduce the number of available training samples, possibly to zero for small training regions.

Overlapping tiles, i.e., partly re-using data, may alleviate this problem, at the risk of producing overfitted models that fail to generalise well. This can be observed in figure 4(d), for landclass 7: the training region is smaller than the tile size, which means that without overlap, there would be no training samples at all. With a constant stride size, the overlap is 97.5% and produces 519 training samples. Yet, accuracy for this class is only 33%, due to lack of independent training samples. In general, classes with small training regions yield the lowest class accuracy (i.e. classes 2, 5 and 7 in our example). In figure 4(a), the shaded region indicates the range of per-class accuracy. The maximum per-class accuracy is always close to 100 i.e., there is always a class that is almost perfectly predicted. The minimum per-class accuracy however decreases for tile sizes larger than 30 km, quite dramatically so for the support vector models (red and blue dashed lines), and slightly less severe for RF-100 (black). It is this misclassification of small training classes which reduces overall accuracy as tiles get larger—we remind the reader that we define overall accuracy as the mean of post-training class accuracy.

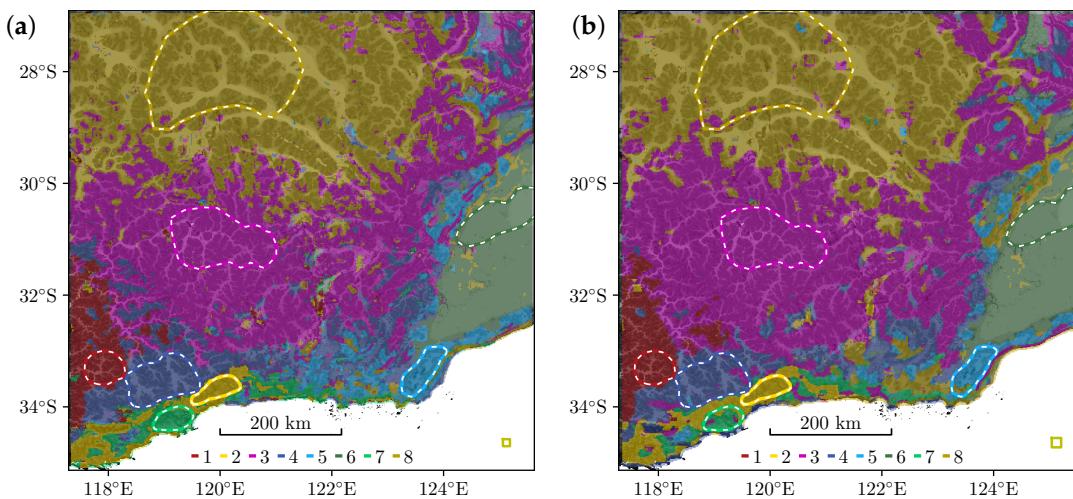


Figure 5. Best maps as produced by the remaining classifiers: (a) SVC-RBF at $s = 13$ km, (b) SVC-LIN at $s = 17$ km.

4.2. Feature importance

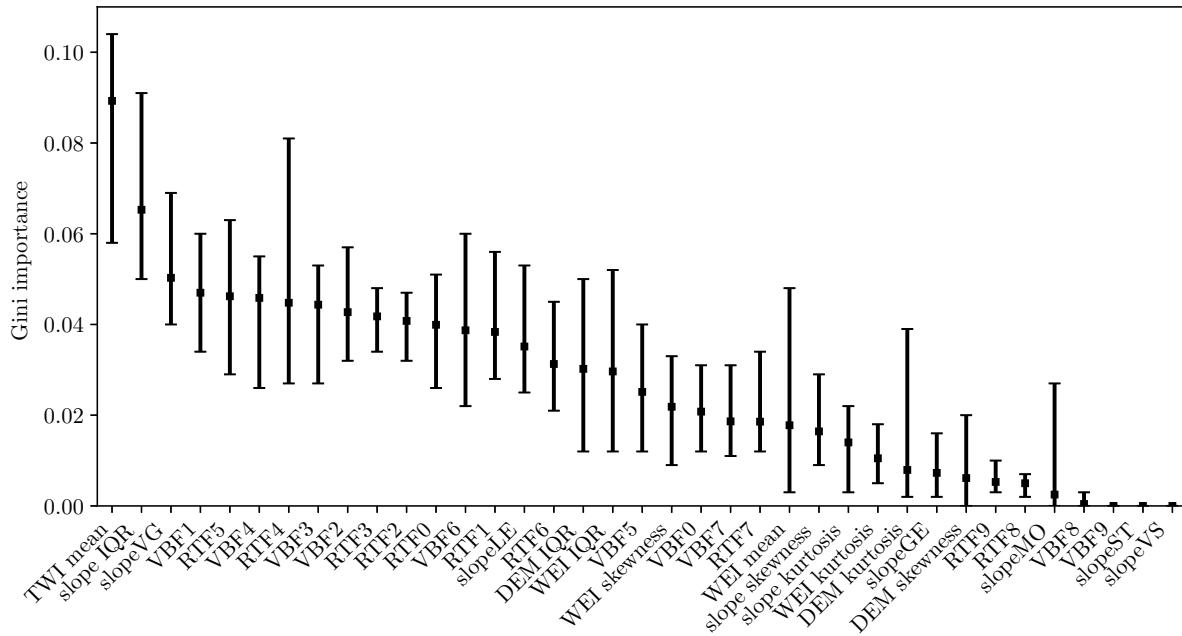


Figure 6. Gini importance for the 37 features, see tables 1 and 2. Squares denote the mean importance, while errorbars show minimum and maximum importance for runs of different tile size and overlap.

Figure 6 shows the Gini importance for the 37 features, averaged over all runs of different tile size and overlap, as discussed in section 4.1. Squares denote the mean importance, while errorbars show minimum and maximum importance. While there is some variation, the overall trend is confirmed: features that are important on average generally remain so irrespective of resolution and tile size. The most important features are topographic wetness index, slope IQR, and slopeVG, with the first two being significantly more important than the rest. Importance for subsequent features decreases approximately linear. The vast majority of features does indeed contribute to the model. Least important ones (VBF9, slopeST, slopeVS) are so because there are very few tiles with these values.

4.3. Circular tiles

Square tiles are easy to implement, but come with two issues. First, the resulting classification is not rotationally invariant: rotating the source data changes which part of said data gets aggregated for a tile (unless we rotate by multiples of 90 degrees). Consequently, the predicted map changes, which is generally unwanted. Second, square tiles tend to produce blocky shapes, most apparent for large block sizes, and especially for RF-100. Circular tiles can alleviate both issues. We also found they improve accuracy slightly (by $\approx 1\%$), at the expense of a similar increase in aggregation time. Arrows in figure 7 highlight some rather subtle differences between square and circular tiles for RF-100 at $s = 43$ km.

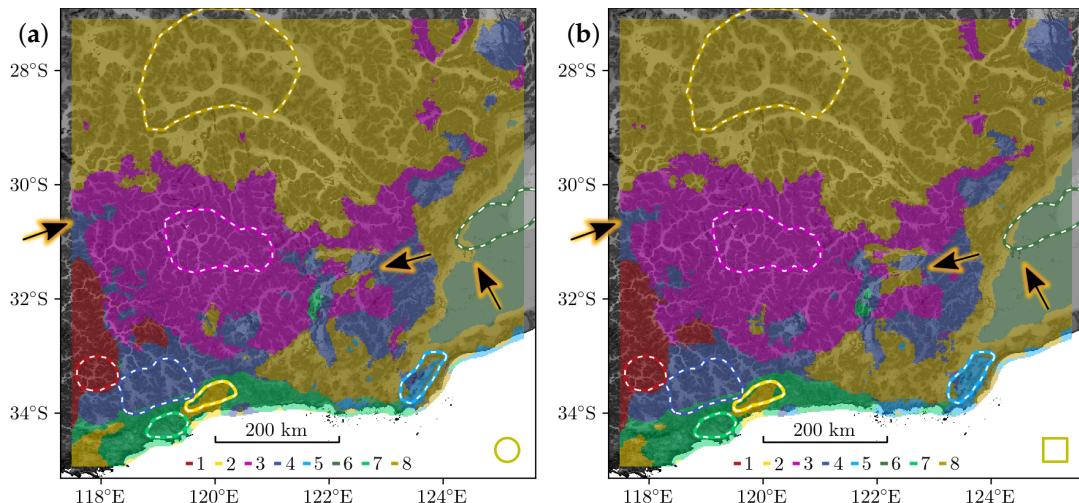


Figure 7. Circular tiles (a) tend to result in less blocky shapes, and slightly improve detail over rectangular tiles (b). Markers indicate some of the generally subtle differences. RF-100 at $s = 43$ km.

4.4. Transition belts

Landscapes vary in morphology based on different climatic conditions, basement architecture and lithologies, tectonic activity and sedimentary dynamic regimes. All these factors have a large footprint. Major changes in any of the variables result in changes in morphology. Often gradual in nature, we refer to these as transition belts.

However so far we have posed the landscape classification problem as binary: a tile belongs to one and only one class. By definition this leads to a sharp boundary between any two classes. This section describes a feature that implements transition belts.

Of mostly cosmetic value would be a purely geometric approach, in which one may define a belt of fixed width (e.g. 20 km) along class boundaries. Support Vector Machines and Random Forests allow for more advanced approaches, each providing suitable but slightly different measures.

Support Vector Machines provide for every sample a distance to the decision boundary. Note that this is a distance in feature space, entirely unrelated to a distance on the map. This distance could be considered a proxy for the probability that a sample belongs to a class; a sample right at the decision boundary is equally likely to belong to either class. Since our features are normalised, we could choose a threshold distance beyond which a sample is considered to belong to a particular class with 100% certainty. Below that threshold we could linearly interpolate the probability.

Random Forests report for every class a probability that a given sample belongs to that class. The probability is simply the portion of Decision Trees in the Random Forest that predict said class for the sample in question.

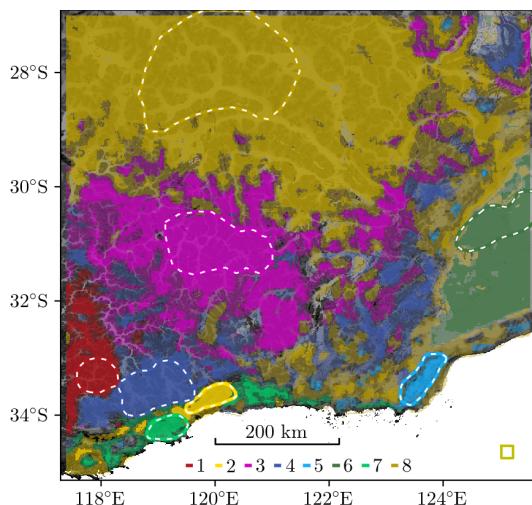


Figure 8. Map with transition belts, RF-100 at $s = 21$ km. Clear, semi-opaque, and mostly opaque areas denote < 40%, 40-80%, and > 80% probability, respectively, that a given tile belongs to the indicated class.

This is the idea we have implemented, results of which we show in figure 8. Transparency represents probability or uncertainty, with opaque areas corresponding to a high probability (> 80%) that a given sample belongs to the class indicated by the colour. Mostly translucent areas therefore indicate a slow, gradual transition from one class to another and appear as rather wide belts between classes. Conversely, a narrow transition belt indicates a distinct, sharp class boundary. This is purely a post-processing feature and comes with no significant computational costs.

4.5. Novelty detection

Located in the SW of our model region, the Stirling Range is very distinct from its surroundings, rising to ≈ 1100 m above sea-level in an otherwise largely flat area. A domain expert would clearly identify it as a separate class. However, we have omitted the Stirling Range from our classification so far, as its rather small spatial extent provides very few training samples at our 3 arc sec DEM resolution. For the same reason we have omitted coastal regions, distinct from the inland patterns due to the modern sedimentary dynamics of the coast (e.g., erosional regime of the waves, coastline and mobile sand dune systems).

It would be useful if our algorithm detected such instances, i.e. those which do not resemble any training data. Known as outlier or novelty detection, a number of algorithms exists for this task, of which we have tested OneClassSVM, IsolationForests and LocalOutlierFactor. Technically, we combine training data of all classes into a single class and train a model. We then predict whether a new sample belongs to that single class (inlier) or not (outlier/novelty). Implementation is straightforward and the required additional training is cheap.

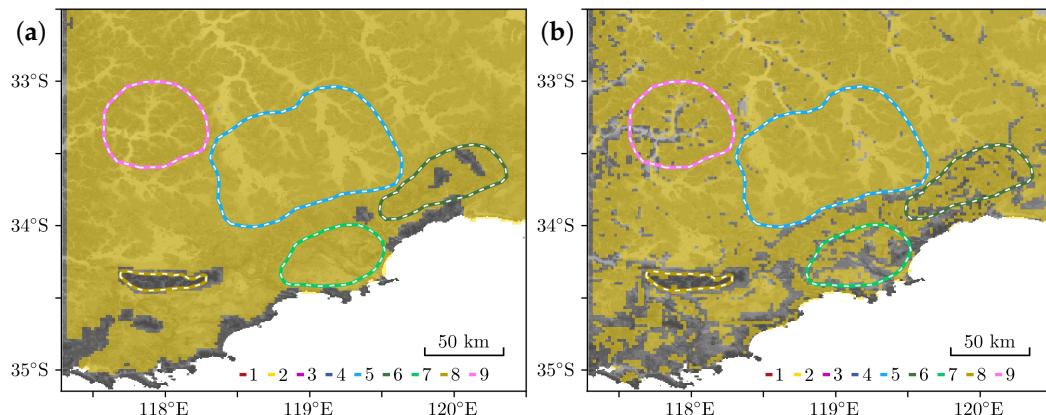


Figure 9. Novelty detection, zoom of the SW of our model region. (a) IsolationForest, $s = 8.7 \text{ km}$. The detected region matches the Stirling Range as marked by a human domain expert. LocalOutlierFactor (b) produces many false positives.

Example results are shown in figure 9. In panel (a), the IsolationForest classifier clearly identifies as unknown (i) the Stirling Range, (ii) a similar region SW to it, and (iii) some costal regions. Almost everything else is classified as known.

This classifier detects the Stirling Range at all tile sizes we have tested, with smaller tile sizes generally resulting in more detailed shapes which better match the boundary drawn by the domain expert. OneClassSVM (not shown) performs about the same up to $s = 17 \text{ km}$, beyond which results degrade. LocalOutlierFactor (figure 9b) produces too many false positives to be useful at all tile sizes we have tested.

5. Discussion and conclusion

We have presented an automatic landscape mapping methodology. For every landclass present in the test region, a domain expert defines a sample region. After training on these sample regions, Machine Learning algorithms are able to predict a dense map. We tested our Python implementation using data derived from a 3 arc second Digital Elevation Model for a region in Western Australia of approximately $1000 \times 1000 \text{ km}^2$. At an optimised tile size we achieve an accuracy of almost 98%, with a reasonable runtime of a 48 minutes.

Crucial to classification accuracy is spatial aggregation of the raw pixel values, but in a way that conserves some spatial variation. We found a per-tile normalised histogram approach to work well. Classification based on raw pixel values fared much worse, which is consistent with the observation that performance of the normalised histogram approach degrades as the tile size approaches zero, see figure 4. While currently the most computationally expensive, this aggregation stage could see a significant speedup through advanced vectorisation, or possibly a C++ implementation.

We also presented three extensions: (i) circular tiles, which enable rotationally invariant classification, (ii) transition belts, which indicate how gradual or abrupt landscapes change, and (iii) novelty detection, which can identify regions which do not match any of the landclasses defined during training.

The methodology presented is a building block of a future toolbox that will allow to map landscape types at continental scale using remote sensing techniques and machine learning algorithms. The approach is scalable and repeatable, both spatially and temporally. We note however that ML does not necessarily remove subjectivity. Results still depend on which and how many polygons the domain expert chooses for training, and the size and stride of tiles. Larger tiles produce a more contiguous but lower resolution map. Conversely, smaller tiles result in a more detailed, and at some point, noisy

map. The choice of ML algorithm is not overly critical: Random Forest and Support Vector Machine classifiers generate quite similar maps, which is a favourable outcome.

This work builds on previous studies about characterisation of landscapes in the South of Western Australia and world-wide [e. g. 4,5,21, and references therein]. The diversity and distribution of landscapes mapped in this study support previous results presented by domain-experts, which were based on field observations [7]. Said study identified four main landscape domains in several field regional transects, conducted at different latitudes in the South of Western Australia (Figure 1). These transects were designed to capture the geomorphological variability between typical landscape features in the Yilgarn Craton to the coastline, in order to characterize the landscape evolution.

The landscape changes were summarized from the topographically high, dissected environment with thick cover development and uneven basement topography, to the nearly flat coastal regions with thin cover and sand dune systems at the coastline. These differential landscape domains were identified in a previous study and classified in a framework of four different landscape types: (i) Albany (green, figure 9); (ii) Kalgoorlie-Norseman (pink and dark blue); (iii) Esperance (bright blue and dark green); and (iv) Neale (dark yellow) [9].

Gonzalez-Alvarez et al. [7] faced three main challenges after reconstructing the transects and the observations from the field: (i) how to infer their direct observations along the traverses laterally, (ii) how to clearly identify boundaries between landscape types along the traverses, and (iii) how to profile with specific features or combination of features the landscape types. The present study identifies the four landscape domains mentioned above, which substantiates the methodology. It also addresses all three of the above challenges. In addition, our results are geologically meaningful since they are consistent with the geological trends and main basement features in the region. The extension and boundaries of the landscape types defined are an important contribution to the understanding of landscape evolution of this region in Western Australia.

In the context of mineral exploration in regions of low-relief and intensely weathered, our results support that DEM can be used to (i) discriminate and characterise landscape variability types at large scale, (ii) map the extension of the landscape domains, (iii) quantify their surface geometrical features, and (iv) establish the transitional boundaries between landscapes. These outputs are of real help to design and interpret surface geochemical surveys at a regional scale in areas of low-relief and intensely weathered landscapes. The variability of the landscape types can reflect geological, tectonic, and stratigraphic changes. Identifying such changes can be fundamental for a more efficient interpretation of surface geochemical data (e. g. detecting the geochemical footprint of potential ore deposits at depth, establishing changes in surface biodiversity associated to geological changes at depth, detecting diverse tectonic regimes, etc.).

Acknowledgements

We would like to thank Greg Smith for his insights and work when the project started, and Tania Ibrahimi for her assistance with GIS and the preparation of some of the figures. Support by the Discovery Program through its strategic funding (CSIRO, Mineral Resources) is gratefully acknowledged.

Author Contributions: methodology, software, validation, formal analysis, investigation, data curation, writing—original draft preparation, visualization, T.A.; conceptualization, writing—review and editing, project administration, funding acquisition, I.G.-A., J.K.

Funding: Support by the Discovery Program through its strategic funding (CSIRO, Mineral Resources) is gratefully acknowledged.

Acknowledgments: We would like to thank Greg Smith for his insights and work when the project started, and Tania Ibrahimi for her assistance with GIS and the preparation of some of the figures.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

DEM	Digital Elevation Model
ML	Machine Learning
RF-100	Random Forest consisting of 100 decision trees
SRTM	Shuttle Radar Topography Mission
SVC-LIN	Support Vector Machine classifier with linear kernel
SVC-RBF	Support Vector Machine classifier with radial basis function kernel

References

1. González-Álvarez, I.; Goncalves, M.; Carranza, J. Introduction to the Special Issue- Challenges for Mineral Exploration in the 21st Century: Targeting Mineral Deposits Undercover. *Ore Geology Reviews Special Issue* **2020**, *126*, 103785.
2. Summerfield, M.A. Neotectonics and landform genesis. *Progress in Physical Geography* **1987**, *11*, 384–397.
3. Ollier, C.D. Evolution of the Australian landscape. *Marine and Freshwater Research* **2001**, *52*, 13–23.
4. Goudie, A. *Encyclopedia of geomorphology*; Routledge, 2014.
5. Twidale, C.R. *Ancient Australian Landscapes*; Rosenberg Pub Pty Limited, 2007.
6. Pain, C.F.; Pillans, B.J.; Roach, I.C.; Worrall, L.; Wilford, J.R. Old, flat and red—Australia's distinctive landscape. *Shaping a nation: A geology of Australia* **2012**, pp. 227–276.
7. González-Álvarez, I.; Salama, W.; Anand, R.R. Sea-level changes and buried islands in a complex coastal palaeolandscape in the South of Western Australia: Implications for greenfield mineral exploration. *Ore Geology Reviews* **2016**, *73*, 475–499.
8. Butt, C.R.M.; Lintern, M.; Anand, R.R. Evolution of regoliths and landscapes in deeply weathered terrain—implications for geochemical exploration. *Ore Geology Reviews* **2000**, *16*, 167–183.
9. Anand, R.R.; Butt, C.R.M. A guide for mineral exploration through the regolith in the Yilgarn Craton, Western Australia. *Australian Journal of Earth Sciences* **2010**, *57*, 1015–1114.
10. González-Álvarez, I.; Boni, M.; Anand, R.R. Mineral exploration in regolith-dominated terrains: Global considerations and challenges, 2016.
11. Eglington, R.A., Ed. *The Regolith Glossary: Surficial Geology, Soils and Landscapes*; CRC LEME: Perth, 2001.
12. Wyborn, L.A.I.; Heinrich, C.A.; Jacques, A.L. Australian Proterozoic mineral systems: essential ingredients and mappable criteria. AusIMM, 1994, pp. 109–115.
13. Otto, J.C.; Prasicek, G.; Blöthe, J.; Schrott, L., GIS Applications in Geomorphology. In *Comprehensive Geographic Information Systems*; Huang, B., Ed.; Elsevier, 2018; pp. 81–111. doi:10.1016/B978-0-12-409548-9.10029-6.
14. Costanza, J.K.; Riitters, K.; Vogt, P.; Wickham, J. Describing and analyzing landscape patterns: where are we now, and where are we going? *Landscape Ecol* **2019**, *34*, 2049–2055.
15. Gustafson, E.J. How has the state-of-the-art for quantification of landscape pattern advanced in the twenty-first century? *Landscape Ecology* **2019**, *34*, 2065–2072.
16. Frazier, A.E.; Kedron, P. Landscape metrics: past progress and future directions. *Current Landscape Ecology Reports* **2017**, *2*, 63–72.
17. Lausch, A.; Blaschke, T.; Haase, D.; Herzog, F.; Syrbe, R.U.; Tischendorf, L.; Walz, U. Understanding and quantifying landscape structure—A review on relevant process characteristics, data models and landscape metrics. *Ecological Modelling* **2015**, *295*, 31–41.
18. Jasiewicz, J.; Niesterowicz, J.; Stepinski, T. Multi-resolution, pattern-based segmentation of very large raster datasets. International conference on GIScience short paper proceedings, 2016, Vol. 1.
19. Jasiewicz, J.; Netzel, P.; Stepinski, T.F. Landscape similarity, retrieval, and machine mapping of physiographic units. *Geomorphology* **2014**, *221*, 104–112.
20. Jasiewicz, J.; Stepinski, T.; Niesterowicz, J. Multi-scale segmentation algorithm for pattern-based partitioning of large categorical rasters. *Computers & Geosciences* **2018**, *118*, 122–130.
21. Ollier, C.D., Applied Geomorphology; Elsevier: Amsterdam, 1978; chapter Terrain classification: Methods, Applications and Principles.

22. Wilford, J.R.; Searle, R.; Thomas, M.; Pagendam, D.; Grundy, M.J. A regolith depth map of the Australian continent. *Geoderma* **2016**, *266*, 1–13.
23. de Caritat, P.; Main, P.T.; Grunsky, E.C.; Mann, A.W. Recognition of geochemical footprints of mineral systems in the regolith at regional to continental scales. *Australian Journal of Earth Sciences* **2017**, *64*, 1033–1043.
24. Caruso, A.; Clarke, K.; Tiddy, C.; Delean, S.; Lewis, M. Objective Regolith-Landform Mapping in a Regolith Dominated Terrain to Inform Mineral Exploration. *Geosciences* **2018**, *8*, 318.
25. Mercader, J.; Martí, R.; González, I.J.; Sánchez, A.; García, P. Archaeological site formation in rain forests: insights from the Ituri rock shelters, Congo. *Journal of Archaeological Science* **2003**, *30*, 45–65.
26. Abdi, A.M. Land cover and land use classification performance of machine learning algorithms in a boreal landscape using Sentinel-2 data. *GIScience & Remote Sensing* **2020**, *57*, 1–20, [<https://doi.org/10.1080/15481603.2019.1650447>]. doi:10.1080/15481603.2019.1650447.
27. Zhang, Y.; Li, Q.; Huang, H.; Wu, W.; Du, X.; Wang, H. The combined use of remote sensing and social sensing data in fine-grained urban land use mapping: A case study in Beijing, China. *Remote Sensing* **2017**, *9*, 865.
28. Lidberg, W.; Nilsson, M.; Ågren, A. Using machine learning to generate high-resolution wet area maps for planning forest management: A study in a boreal forest landscape. *Ambio* **2019**, pp. 1–12.
29. Maxwell, A.E.; Warner, T.A.; Fang, F. Implementation of machine-learning classification in remote sensing: an applied review. *International Journal of Remote Sensing* **2018**, *39*, 2784–2817, [<https://doi.org/10.1080/01431161.2018.1433343>]. doi:10.1080/01431161.2018.1433343.
30. Jasiewicz, J.; Netzel, P.; Stepinski, T. GeoPAT: A toolbox for pattern-based information retrieval from large geospatial databases. *Computers & Geosciences* **2015**, *80*, 62–73.
31. Gallant, J.C.; Wilson, N.; Dowling, T.I.; Read, A.; Inskeep, C. SRTM-derived 3 Second Digital Elevation Models Version 1.0. Technical report, Geoscience Australia, 2009.
32. Gallant, J.C.; Dowling, T.I. A multiresolution index of valley bottom flatness for mapping depositional areas. *Water resources research* **2003**, *39*.
33. Kuhn, M.; Johnson, K. *Applied predictive modeling*; Springer, 2013.
34. Fan, R.E.; Chang, K.W.; Hsieh, C.J.; Wang, X.R.; Lin, C.J. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research* **2008**, *9*, 1871–1874.
35. Chang, C.C.; Lin, C.J. LIBSVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 27:1–27:27. doi:10.1145/1961189.1961199.
36. Breiman, L. *Classification and regression trees*; Routledge, 2017.
37. Breiman, L. Random forests. *Machine Learning* **2001**, *45*, 5–32.
38. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.
39. Gallant, J.C.; Dowling, T.I.; Austin, J. Multi-resolution Ridge Top Flatness (MrRTF) v2. Technical report, CSIRO, 2013.
40. Gallant, J.C.; Austin, J. Topographic Wetness Index derived from 1" SRTM DEM-H. v2. Technical report, CSIRO, 2012.
41. Gallant, J.C.; Austin, J. Slope derived from 1" SRTM DEM-S. v4. Technical report, CSIRO, 2012.
42. Böhner, J.; Antonić, O. Land-surface parameters specific to topo-climatology. *Developments in soil science* **2009**, *33*, 195–226.
43. Gerlitz, L.; Conrad, O.; Böhner, J. Large-scale atmospheric forcing and topographic modification of precipitation rates over High Asia—a neural-network-based approach. *Earth System Dynamics* **2015**, *6*, 61.
44. Gallant, J.C.; Austin, J. Slope Relief Classification derived from 1" SRTM DEM-S. v3. Technical report, CSIRO, 2012.
45. Han, J.; Miller, H.J. *Geographic data mining and knowledge discovery*; CRC Press, 2009.