BENCHMARKING DEEP REINFORCEMENT LEARNING ALGORITHMS FOR UNSUPERVISED HYPERSPECTRAL BAND SELECTION

Pratik Aher[1], Romit Barua[1], Daniel Furman[1*]

[1]University of California, Berkeley, CA 94720, USA
{pratikaher88, romit_barua, daniel_furman}@berkeley.edu

ABSTRACT

Unsupervised band selection is an important technique in some applications for processing high-dimensional hyperspectral image datasets. Here, we test the performance of eight deep reinforcement learning agents (four architectures, two reward schemes) for this task on five hyperspectral datasets. Unsupervised band selection is, at heart, a combination of a robust evaluation criteria and an effective band selector. In regards to the latter, we develop four deep reinforcement learning architectures for this task and compare their effectiveness including deep Q-networks, double deep Q-networks, actor critic, and soft actor critic. Only vanilla deep Q-networks have been implemented in the literature for unsupervised band selection to date [Mou et al., 2022]. As in Mou et al., unsupervised band search is cast here as a reinforcement learning problem where the agent's goal is to learn a policy that chooses bands until K bands are reached. It's common to use $K = 30$ in the literature, and we follow convention and only perform experiments with this value hard-coded at 30 across our project. We implement two reward schemes in regards to the evaluation criterion mentioned above: absolute correlation (corr) and normalized mutual information (mi). When an agent takes an action and moves from the current state to the next state, the reward $R(s, s')$ is calculated by the following $R(s, s') = \exp(\text{metric}(s) - \text{metric}(s'))$. As expected, these equations indicate that the reward is positive if the quality of selected bands is improved from state s to state s', since it's advantageous to reduce the metric (multi-collinearitry or mutual information) in a given set. Our contributions to the field here include both the use of mutual information as a metric and including an exponential in the reward formula, both of which improved our results. After running the agent training experiments, we performed evaluation with both the reward metric values (Table 1). and the supervised modeling tasks (Table 3). In the supervised tests, we trained linear classifiers and regressors to compare the effectiveness of the reinforcement learning agents in terms of downstream modeling performance. In the reward metrics tests, we similarly compare the agents, but this time, we do so in terms of correlation and mutual information directly. The results of our benchmark experiments suggest that, overall, actor-critic methods outperform traditional Q-learning methods in correlation minimization. Further, the Q-network struggles to correctly fit to a mutual information reward metric. We find that random band selection often performs comparably to that of RL methods and other unsupervised methods explored in prior literature. The code is publicly available.[2].

Keywords Deep reinforcement learning · Hyperspectral imagery · Unsupervised band selection

[*]All authors have contributed equally

[2]`https://github.com/pratikaher88/HyperSpectralRL_V2/tree/master`

1 Introduction

Many works have applied deep reinforcement learning to computer vision tasks with imagery of low band dimensionality Furuta et al. [2018] Vassilo et al. [2020] Li et al. [2020a] Li et al. [2020b] Rout et al. [2020] Chu et al. [2019] Liao et al. [2019] Anh et al. [2020] Casanova et al. [2020] Uzkent et al. [2019] Liu and Tang [2021] [Yu et al., 2022]. Examples of these types of data include typical RGB color images, multispectral datasets, among many other types. Recently, a couple studies demonstrated applications of deep reinforcement learning in hyperspectral images, which have much larger (often over 100x) band dimensionality [Feng et al., 2022] [Mou et al., 2022]. However, these studies used very small, single-image datasets that are not large enough to mimic applications with hyperspectral datasets collected at scale, e.g., [Hook et al., 2014]. In this project, we aim to contribute to the field by benchmarking a number of deep reinforcement learning algorithms for hyperspectral band selection on a more diverse, representative collection of five hyperspectral datasets.

Hyperspectral data can be defined as imagery with hundreds of stacked, contiguous bands narrowly set apart in electromagnetic wavelength. Sensors can capture data at different ranges, for example, across the near-infrared wavelengths as well as visible spectrum wavelengths. One finds applications of hyperspectral imaging frequently in the field of remote sensing, since it can distinguish land-cover details with a high degree of diagnostic power [Wei et al., 2017]. In remote sensing use cases, downstream modeling tasks with hyperspectral data include land cover classification, anomaly detection, change detection, and target detection and recognition (e.g., greenery detection, urbanization analysis, crop area analysis) [Camps-Valls et al., 2013] [Du et al., 2019] [Zhao et al., 2020] [Roy et al., 2021]. Yet, hyperspectral applications extend beyond remote sensing and more generally into the field of computer vision at large. For example, four fifths of the datasets included in the benchmark experiments below were captured using sensors in the laboratory or outside (i.e., sensors not attached to a satellite or an airbone vehicle). To the best of our knowledge, all previous work on applying reinforcement learning to this problem has been done on classical land cover datasets in the remote sensing community [Feng et al., 2022] [Mou et al., 2022].

This work's novel contributions in to the field include the following:

- The application of several deep reinforcement algorithms to the problem for the first time, such as Actor Critic, Soft Actor Critic, and Double Deep Q-Networks. To the best of our knowledge, only a Deep Q-Network has been applied to unsupervised hyperspectral band selection to date [Mou et al., 2022].
- The development of new reward structures for the problem, including the introduction of a new metric (mutual information) and of new implementations (taking an exponential of the rewards) all of which improved the performance of our agents.
- The use of a diverse set of single- and multi-image hyperspectral datasets. This was done to include more data in our benchmark and to better consider challenges that arise with bigger datasets.

2 Background

Hyperspectral images provide rich spectral-spatial information about the world; however, the existence of noise (certain bands don't contain discrimitive information) and correlation (adjacent bands are often redundant) pose significant practical problems in some settings [Sun and Du, 2019] [You et al., 2022]. For example, a significant issue arises in terms of computational overhead in processing these datasets at scale. Take the "Hyperspectral Infrared Imager" launched by NASA's Jet Propulsion Laboratory [Hook et al., 2014]. It has a continuous data rate of 65 Mb/s and a data volume of 5.2 TB/day. Such large data can hinder fast onboard data processing and be a bottleneck in obtaining real-time signals. This is where unsupervised band selection can be valuable as a simple first step to capture and store data more efficiently given the redundancy characteristic to these types of images, where neighboring bands are so close in wavelength.

As alluded to above, one solution to reducing the volume of hyperspectral data is to simply select a subset of representative bands from the imagery, a process known as band selection. Let any hyperspectral dataset $X = \{x_i\} \in \mathbb{R}^{N \times D}$ be represented by band vectors $B = \{b_j\} \in \mathbb{R}^{D \times N}$ where x_i refers to the ith pixel's spectral signature, D is the number of pixels, N is the numbers of bands with $N \ll D$, and b_j corresponding to the jth band. To account for multi-image datasets, a pre-processing step is assumed where one vertically stacks individual images per band, and so the total number of pixels D is equivalent to the total number of pixels across all individual images in the dataset. Band selection is the process of choosing m representative bands $M = B(:, x) \in \mathbb{R}^{D \times m}$.

As opposed to feature extraction, which takes a linear or non-linear transformations of images, band selection is particularly well suited for situations when modeling practitioners want physically meaningful data, for example, when interpretability is a main objective [Sun and Du, 2019] [Mou et al., 2022]. With so much redundancy amongst channels,

the interpretability of downstream supervised models and exploratory analytics becomes difficult on hyperspectral data. This difficulty is primarily caused by both the multi-collinearity between features and by the shear number of features. For example, when using SHAP values for interpretation of an XGBoost classifier, reducing the number of bands from 300 to 30 can provide a more practical lens through which to assess feature importance. The key assumption one makes when conducting band selection is that the retained bands preserve the spectral information of relevance to the given use case or application.

As follows, unsupervised band selection can be viewed as a simple combinatorial optimization problem. One identifies a cluster of representative bands and disregards the rest, which are deemed non-informative. Consider a hyperspectral image with 300 bands. Say our goal is to select 30 non-redundant bands within. The number of possible combinations is $\binom{300}{30}$, roughly 2×10^{41} options. The magnitude of this band combinations space makes brute-force methods impractical and motivates the need for some type of heuristic or learning selector.

3 Methodology

3.1 Band Selection as a Reinforcement Learning Problem

Unsupervised band selection is formulated as a sequential forward search process akin, in most ways, to [Mou et al., 2022]. The agent is tasked with deciding which band to select and include in the set at each time step. During the procedure, the agents explore this tailored environment via actions and observed rewards and states.

The action of the agent is to choose a spectral band from the image at each time step. Thus, the complete set of actions is simply the given combination of bands in the set. The state is the action history and is denoted by a multi-hot encoded vector that records which spectral bands have been chosen in the past. The next state is encoded by a transition function as a possible outcome of taking an action at a state. The next state is deterministically specified for each state-action pair. Reward functions are developed below to measure how much the agent improves from state s to s'.

We instantiate two reward schemes here: absolute correlation (corr) and normalized mutual information (mi). When an agent takes an action and moves from the current state to the next state, the reward $R(s, s')$ is calculated by the following $R(s, s') = \text{exp}(\text{corr}(s) - \text{corr}(s'))$ or $R(s, s') = \text{exp}(\text{mi}(s) - \text{mi}(s'))$. Intuitively, these equations indicate that the reward is positive if the quality of selected bands is improved from state s to state s'. This is because both absolute correlation and normalized mutual information encode, from 0 to 1, the magnitude of redundancy amongst a set of bands (i.e., it's advantageous to reduce the multi-collinearitry / mutual information in a given set of bands). The exponentiation of the difference was included to grow larger values more and, ultimately, help the agents iterate towards a better solution.

- Pearson correlation. The Pearson correlation (corr) coefficient measures the linear relationship between pairs of data. Like other correlation coefficients, this one varies between -1 and +1 with 0 implying no correlation. Correlations of -1 or +1 imply an exact linear relationship. Positive correlations imply that as x increases, so does y. Negative correlations imply that as x increases, y decreases. Here, we take the absolute value of the Pearson correlation because we are solely interested on obtaining a 0 to 1 scale of redundancy regardless of the positivity or negativity of the association. We employed the scipy stats function pearsonr here (link). To calculate the metric across a band set of size K, we record the absolute correlation for each band pair except for those along the diagonal (where corr = 1) and divide by K^2.
- Normalized mutual information. The mutual information is a measure of the similarity between two instances of the same data. In other words, it's the distance between two probability distributions. Normalized Mutual Information (referred to here as 'mi') is a normalization of the mutual information score to scale the results between 0 (no mutual information) and 1 (entirely mutual information). One can think of mutual information is a measure of both non-linear and linear associations between pairs of data, whereas, correlation is only concerned with linear relationships. We employed the sklearn function of mi here (link). This was particularly helpful because the sklearn implementation estimated the probability distributions involved for us. To calculate the metric across a band set of size K, we record the mi for each band pair except for those along the diagonal (where mi = 1) and divide by K^2.

3.2 Deep RL for Optimal Band Selection

3.2.1 Q-network architecture

For hyperband selection, we first implemented offline-versions of vanilla DQN [Watkins, 1989] and double DQN (DDQN) [van Hasselt et al., 2015]. Beyond replicating the work of Mou , we felt that DQN was a to attempt first for

two key reasons. First, DQN has relatively few hyper-parameters to tune. Second, given the single network architecture, we believed that DQN was a good way to better understand how the Q-Network learns the "reward to go" from the two unique reward functions. In this architecture, we first populated a replay buffer with trajectories and sampled from the buffer to train our Q-Network to learn the "reward-to-go" based on either the correlation or mutual information metrics. Our Q-Network included an input layer taking in the state and two linear layers separated by the ReLU activation functions. Each layer was the size number of total bands in the image * 2. The output, representing the "reward-to-go", returned an array size of the number of total hyperspectral bands.

In addition to DQN, we implemented DDQN. One of the major weaknesses of DQN is the risk of overestimating the Q-value. The Q-value estimation is a function of the states that have been visited and the prior actions taken when at those states. To account for this potential issue, double DQN separates the action selection from the loss function target setting. By using the DQN Network to select actions and the target network to determine the estimate of the Q-value because of that action, we can avoid overestimating the Q-values.

3.2.2 Actor Critic architecture

We also implemented the Actor-Critic algorithm [Sutton et al., 2000]. Unlike DQN, which uses a deep neural network to determine the Q-value for each action in a given state and selects the action with the highest predicted Q-value, Actor-Critic has an additional network, the actor, to select actions. The actor is updated using V-value predictions from the critic network.

The actor network, which is used to select actions, has two layers of size 64 and uses a softmax activation function. Given a state as input, the network outputs an action distribution from which the action is sampled. On the other hand, the critic network has two layers of size 64 and uses a linear activation function. It takes in the current state as input and outputs the V-value for that state.

During learning, the advantage of selecting a specific action in a given state is calculated as the difference between the predicted Q-value for the current state and the predicted Q-value for the next state. This advantage value is then used to update the policy distribution in the direction suggested by the critic network. Both the actor and critic parameters are updated on each learning step, with the actor being updated using policy gradients and the advantage value, and the critic being updated by minimizing the mean squared error using the Bellman update equation.

Additionally, we implemented Soft-Actor Critic (SAC) [Haarnoja et al., 2018], an off-policy actor-critic deep reinforcement learning algorithm based on the maximum entropy reinforcement learning framework. Unlike traditional reinforcement learning algorithms that only aim to maximize rewards, SAC introduces a entropy regularizer. In our case, since we are dealing with a discrete action space, the actor network outputs a probability mass function instead of a density function. The critic network in SAC takes a state and action as input and outputs the Q-value for that state-action pair. To account for the fact that the Q-values can be overestimated when using gradient-based optimization, SAC uses a technique called double Q-learning, in which two Q-networks are trained and the smaller of the two predicted Q-values is used.

3.2.3 Implementation Details and Associated Learnings

Given that Mou's code is not open source, we had to generate the band selection environment and integrate the Reinforcement Learning algorithms from scratch. This implementation introduced a number of unexpected challenges including issues with the reward function, difficulty getting the Q-network to correctly learn the "reward-to-go", inferior chosen actions and very high compute times.

During the initial implementation of the DQN algorithm, we found that even after considerable training, the Q-network had a very tough time differentiating between the Q-values of different bands. Often times, we would see that the model would select the same band over and over again. Given that our reward function was correlation, our initial intuition was that given the same band will have a correlation of 1 when calculated against itself, the network itself will learn to avoid selecting a band that has already been selected. In practice, we found that the Q-network was unable to have drastically different Q-values for bands that have already been selected vs. those that have not been selected. To further test our hypothesis, we gave a network a very bad hard-coded correlation. This just caused all of the predicted Q-values to "blow up" early in the training process, indicating instability in the network. Eventually to fix this issue, we made two changes. First, we took the exponential of the step reward. This non-linear transform of the reward function helped the model differentiate between rewards, and thereby allowed the network to better pick actions from predicted Q-values. Second, we added logic to the query to prohibit the model from selecting the same band twice for a given trajectory. After making these two changes, we saw the Q-network improved in terms of selecting good and bad bands.

Another issue we dealt with was that after training for a short time, the model select the same 30 bands continuously, in most instances at non-optimal reward values. This is likely due to the fact that the Q-network's predictions are not changing much, so an "ArgMax" model is making the same decisions over and over again. To address this issue, we forced the model to randomly select the 1st band. This forced the model to see more states gain a better understanding of the band selection space. After making this change, the model took much longer before settling on its "favorite bands".

Finally, given the size of the datasets, we found that the calculations for correlation and mutual information were extremely expensive and time consuming. To address this issue, we generated a caching system that saved down the reward between two bands after its first calculation. In future iterations, the model would first search the cache to see if any band pairs had been calculated before and only calculate the pairs that are not found in the cache.

3.2.4 Currently Ongoing Work

In addition to the algorithms highlighted above, we have made some progress on implementing exploration/exploitation based algorithms. Based on the learning discussed above, much of hyperspectral band selection using reinforcement methods is an exploration problem. One of the major issues that we faced in DQN implementation was that the model would make the same decision over and over again and not allow for it to understand the search space effectively. This resulted in us using large random selection parameters to force the model into search. One way to potentially get around this issue is to incorporate a exploration/exploitation structure. The current implementation for Random Network Distillation (RND) can be found on our associated Github. While the code currently compiles trains the model, we are not currently getting expected results and have therefore not included the results in this paper. We will continue to work on the RND implementation and present findings in future work. Currently implementation is publicly available (link).

4 Experiments and Analysis

4.1 Hyperspectral Dataset Descriptions

The five different datasets included each have unique characteristics in terms of supervised task, number of images, size of images, types of labels, range of wavelengths, and sensor type / application (remote sensing, in the laboratory, or outside in the real-world). The oldest was captured in the 1980s while the most recent was published in 2022.

4.1.1 Plastic Flakes Segmentation

The Plastic Flakes dataset was introduced in the paper "Hyperspectral Imaging for Overlapping Plastic Flakes Segmentation" presented at ICIP 2022 [Martinez et al., 2022]. It was captured in a laboratory using a sensor with sensitivity across the near-infrared region of the electromagnetic spectrum. The shape of the imagery is (11, 112128, 224), where the first index is the number of images, the second index is the number of pixels, and the last index is the number of bands. The following sections will follow this shape convention. The original non-flattened images were 876 pixels in height 128 pixels in width (See Figure 1, upper right). The dataset contains the following classes: background, three primary plastics (PP, PE, PET) and four combinations (PP+PE, PP+PET, PE+PET, PP+PE+PET), resulting in eight classes. The task can be thought of as an instance segmentation problem wherein one classifies the plastic flakes in an image by type.

4.1.2 Salient Objects Detection

The Salient Objects dataset was introduced in the paper "Hyperspectral Image Dataset for Benchmarking on Salient Object Detection" presented at QoMEX 2018 [Imamoglu et al., 2018]. It was captured out in the real world using a sensor with sensitivity across the visible region. The shape of the imagery is (60, 786432, 81). The original non-flattened images were 768 pixels in height and 1024 pixels in width (See Figure 1, upper left). The dataset contains the following classes: background, salient object, resulting in two classes. The task can be thought of as an instance segmentation task wherein the aim is to identify objects or regions more attentive than the surrounding areas in a given image.

4.1.3 Indian Pines Land Cover Classification

The Indian Pines dataset is a classical datasaet in the hyperspectral imaging literature and is provided through Purdue University's MultiSpec lab. It was captured from an airborne vehicle operated by JPL (called AVRIS) with sensitivity across the visible and near-infrared regions. The shape of the imagery is (1, 10249, 200). The original non-flattened images were 145 pixels in height and 145 pixels in width with almost half representing an NaN class (See Figure 1, lower left). The task is an agriculture type land classification. The 16 classes are Alfalfa, Corn-notill, Corn-mintill

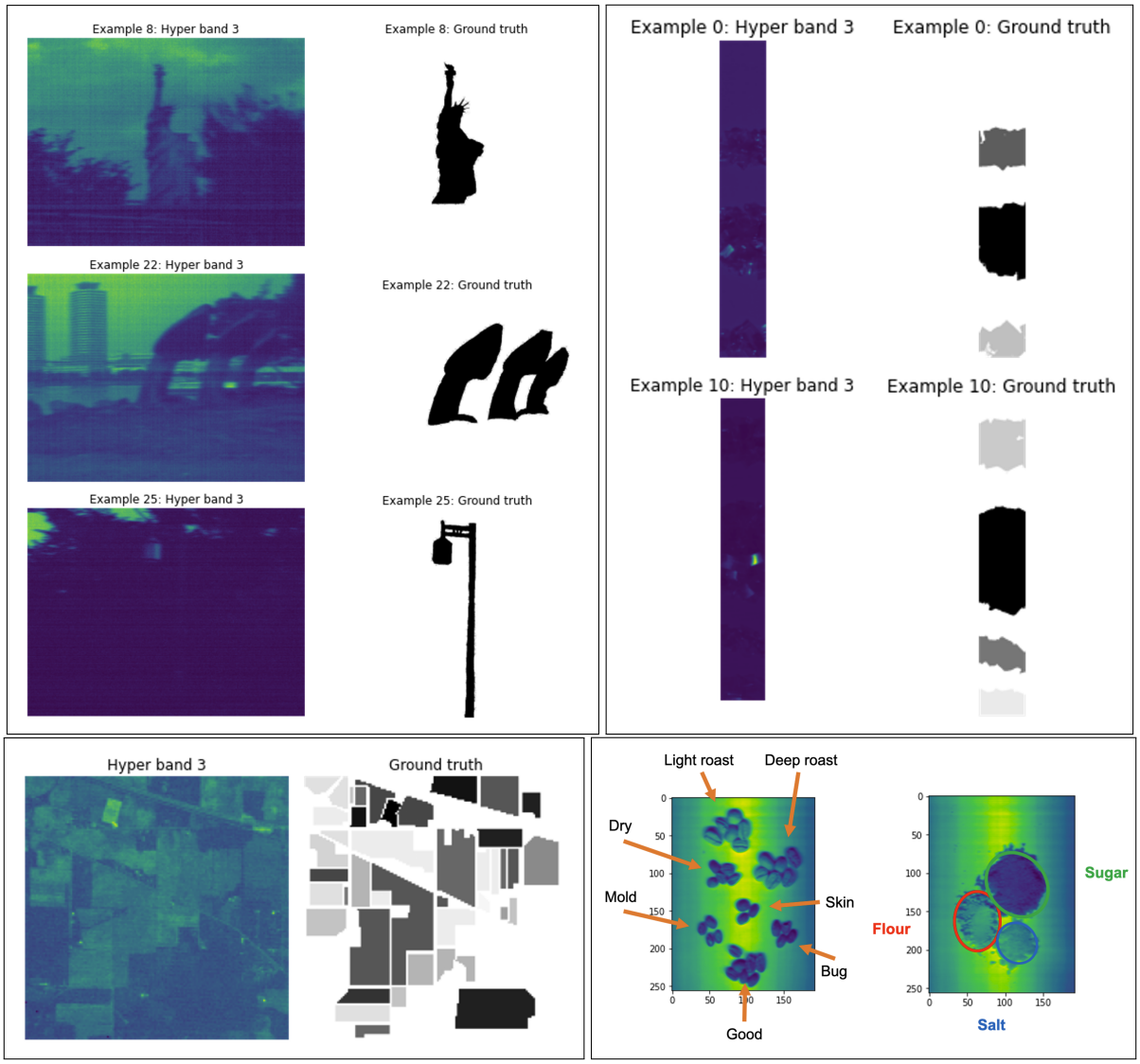

Figure 1: Four datasets included in the benchmark below, as visualized by taking a single random band amongst the hyperspectral imagery. All datasets included in the study are open-access, see dataset citations section for more information. The salient objects dataset sits in the upper left [Imamoglu et al., 2018], an instance segmentation task wherein the aim is to identify objects or regions more attentive than the surrounding areas. In the upper right is the Plastic Flakes dataset [Martinez et al., 2022] for plastic flake type classification. Next, the Indian Pines dataset sits in the bottom left, a classical land classification dataset used across the literature. Next, the coffee and pantries classification images from the foods dataset. The fifth dataset in the benchmark, the soil moisture regression dataset comes loaded as a flattened image and therefore wasn't visualized here.

830, Corn, Grass-pasture, Grass-trees, Grass-pasture-mowed, Hay-windrowed, Oats, Soybean-notill, Soybean-mintill, Soybean-clean, Wheat, Woods, Buildings-Grass-Trees-Drives, Stone-Steel-Towers. We use the 'corrected' version of the dataset employed in [Roy et al., 2021].

4.1.4 Soil Moisture Regression

The Soil Moisture dataset is provided through Kaggle.com. It was measured during a five-day field campaign in May 2017 in Karlsruhe, Germany. An undisturbed soil sample is the centerpiece of the measurement setup, which was captured by a hyperspectral sensor that spans across the visible and near-infrared regions. Unlike the other datasets, this data is reported with band wavelengths stacked in increments of 4; therefore, they cover a larger portion of the spectrum with less bands (and perhaps less adjacent band redundancy). The soil sample consists of bare soil without any vegetation and was taken in the area near Waldbronn, Germany. The task is to predict the percentage of soil moisture, a regression problem. The shape of the imagery is (1, 679, 125). The sample with the driest (minimum) soil moisture is 25.5 and the wettest (maximum) soil moisture is 42.5 [Roy et al., 2021].

4.1.5 Foods Classification

The Foods Classification dataset is provided through Kaggle.com. It was captured (presumably in a kitchen, or perhaps in a laboratory) using a sensor with sensitivity across the near-infrared range. The shape of the imagery is (4, 49152, 96). The original non-flattened images were 256 pixels in height and 192 pixels in width. This dataset comes unlabeled beyond including some images with arrows pointing to the given foods. So, we extracted rectangular patches (of size 20 x 20) from the different images in the areas where the author indicated a certain class being (See Figure 1 for two examples). We ended up with a food type classification for 6 classes with 400 pixels per class. The 6 classes are yatsuhashu, coffee, rice, flour, salt, and sugar.

4.2 Experimental Setting

The key goal of our project was to determine a generalizable approach to select hyperspectral bands that will allow users to both run specific downstream tasks such as classification and regression and store the information for future use. As such, we tested our selected bands across two dimensions: the RL reward and downstream classification tasks.

4.2.1 Non-RL Band Selection Methods

We benchmarked the performance of the different reinforcement learning models against random selection and variance-based ranking methods. To compare against the reward metrics, we randomly selected 50 sets of 30 hyperspectral bands calculated the distribution of correlation and mutual information (See Figure 3). We also implemented variance-based ranking band selection. However, we found that performance using such methods was consistently worse than random band selection. As such, we did not include the variance-based ranking methods in the tables.

4.2.2 Band Selection Methods

To collect band from the reinforcement learning models, we ran each algorithm for 2000 iterations. Given that vastness of the search space, we started the epsilon at 0.99, with a step-wise decay of 0.9999. This resulted in predominantly random selection of bands through the first 250 to 750 iterations. We additionally used a discount term of (gamma) of 0.99. The replay buffer sample size was 10 trajectories with the critic updating 10 times per iteration.

4.2.3 Reward Metrics Downstream Evaluation

While hyperspectral images may be collected with specific tasks in mind, it is reasonable that researchers and practitioners may want to store the data for use in unidentified tasks in the future. As previously mentioned and in-line with other literature in the field, we are using correlation and mutual information as a proxy for information retention.

In order to determine model reward performance, we reviewed the reward function across two dimensions: the reward value for the final set of selected bands (Table 1) and the minimum reward value after iteration 750 (Figure 3, dashed lines). For the latter, we chose to select the minimum value after 750 because that was where we were confident that our model was selecting actions from the policy and not random actions due to the high random selection parameter (epsilon). We have also included the evaluation of downstream reward metrics using all available hyperspectral bands (Table 2). In Table 2, one can see that the full imagery is of much higher redundancy than the RL agents above, in Table 1.

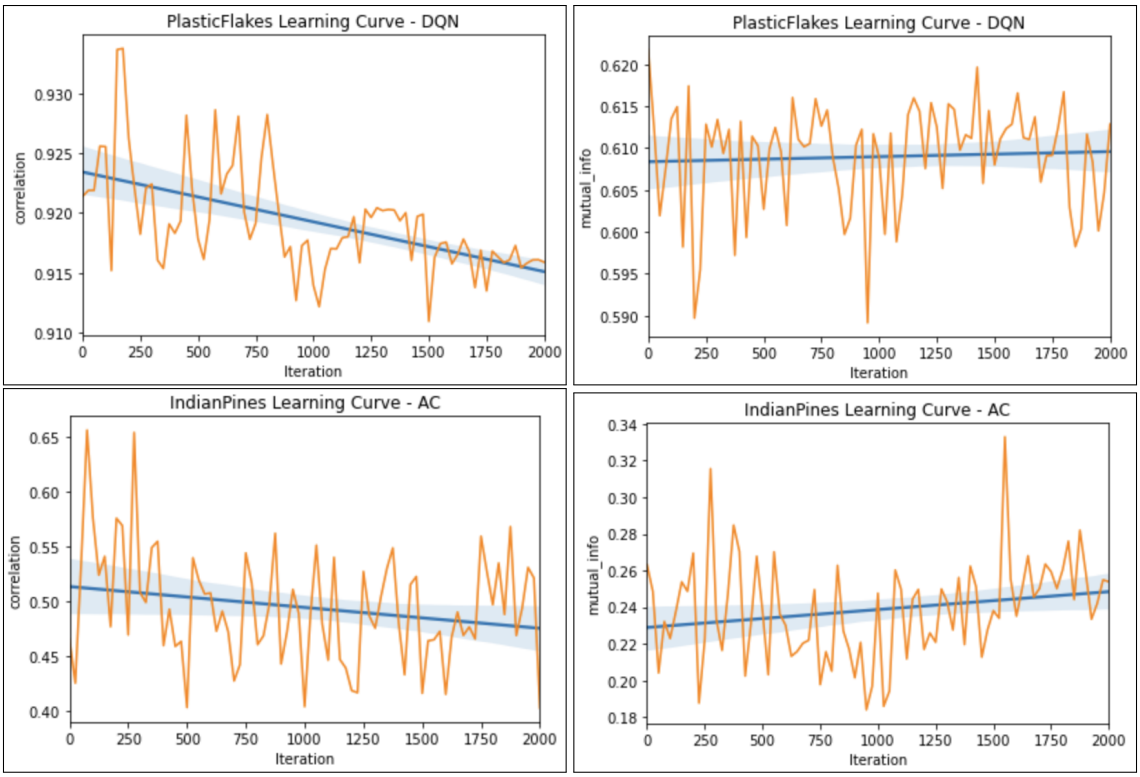

Figure 2: Example learning curves for a correlation task and a mutual information task. Note that we hand selected these learning curves to highlight the key takeaway that the agent is able to learn to minimize the correlation, but struggles to minimize mutual information.

4.2.4 Modeling Downstream Evaluation

To evaluate our selected bands on downstream tasks, we filtered down all the images to the appropriate bands suggested by the reinforcement learning agents or by random band selection. In the case of downstream modeling evaluation, rather running the entire downstream evaluation on 50 sets of randomly selected bands, we only selected a single set. This was primarily to account for the incredibly computationally intensive training, especially when utilizing the multi-image datasets. In the reinforcement learning algorithm and in the reward tests, we had sampled 1% of pixels from Salient Objects and 10% of pixels from Plastic Flakes. This sampling was done in a pre-processing step and then cached so that all team members were using the exact same data sample. For the purpose of downstream modeling evaluation, we utilized all the provided pixels for the multi-image dataset. In the case of multi-image datasets, we trained a separate model per image and averaged the performance across all the images.

For four of the datasets, we ran a Logistic Regression to classify pixels into their appropriate groups. SoilMoisture was a continuous task for which we used Linear Regression. While other more expressive models may have resulted in greater downstream accuracy, we felt that this exercise was not to get the "highest accuracy", but rather determine comparable way, which band selection method is superior for given datasets and tasks. During the classification tasks, we tracked the accuracy, balanced accuracy score and the F1-measure (macro averaged). In the regression task, we used R^2, mean average error and mean squared error. We have also included the evaluation of downstream tasks using all available hyperspectral bands (Table 4).

5 Discussion

5.1 Results

In Table 1, we report the reward metrics for each of the Reinforcement Learning algorithms at the end of training. For all datasets, we find that either Actor-Critic or Soft Actor Critic perform the best in the correlation minimization task. During the course of this project, one of the key takeaways is that training a network to learn the Q-value for correlation

Table 1: Downstream rewards metrics results, selected bands at the end of RL agent training.

Plastic Flakes	corr	mi
DQN-corr	0.9159	**0.5563**
DDQN-corr	0.9355	0.6109
AC-corr	**0.9131**	0.5599
SAC-corr	0.9316	0.6063
DQN-mi	0.9364	0.6128
DDQN-mi	0.9299	0.6034
AC-mi	0.9260	0.6019
SAC-mi	0.9262	0.5995

Salient Objects	corr	mi
DQN-corr	0.0756	0.0693
DDQN-corr	0.0807	0.0793
AC-corr	0.0945	0.0812
SAC-corr	**0.0578**	0.0771
DQN-mi	0.0879	0.0847
DDQN-mi	0.0756	0.0860
AC-mi	0.0707	**0.0683**
SAC-mi	0.0911	0.0818

Soil Moisture	corr	mi
DQN-corr	0.9540	0.6570
DDQN-corr	0.9516	0.6355
AC-corr	0.9558	0.6383
SAC-corr	0.9534	0.6352
DQN-mi	0.9545	0.6407
DDQN-mi	0.9532	**0.6307**
AC-mi	0.9549	0.6484
SAC-mi	**0.9501**	0.6365

Foods	corr	mi
DQN-corr	0.6399	0.6090
DDQN-corr	0.5591	**0.5804**
AC-corr	0.6069	0.5851
SAC-corr	0.5799	0.5896
DQN-mi	0.7098	0.6172
DDQN-mi	0.5903	0.6042
AC-mi	**0.5045**	0.5815
SAC-mi	0.5908	0.6049

Indian Pines	corr	mi
DQN-corr	0.5241	0.2356
DDQN-corr	0.6143	0.2774
AC-corr	**0.4023**	**0.1996**
SAC-corr	0.5162	0.2407
DQN-mi	0.5673	0.2573
DDQN-mi	0.6081	0.3376
AC-mi	0.5394	0.2539
SAC-mi	0.5210	0.2477

Table 2: Downstream rewards metrics results, all bands included.

dataset	corr	mi
Plastic Flakes	0.9549	0.6167
Salient Objects	0.0883	0.0819
Soil Moisture	0.9784	0.6573
Foods	0.5941	0.6066
Indian Pines	0.5566	0.2618

Table 3: Downstream supervised task modeling results, selected bands at the end of RL agent training and one random-set for comparison. The supervised learning classifiers employed were default sklearn logistic and linear regression.

Plastic Flakes	acc	bac	f1
DQN-corr	0.9611	0.9354	0.9353
DDQN-corr	0.9570	0.9271	0.9292
AC-corr	0.9552	0.9251	0.9262
SAC-corr	0.9596	0.9330	0.9330
DQN-mi	0.9568	0.9271	0.9290
DDQN-mi	0.9573	0.9286	0.9287
AC-mi	0.9615	0.9359	0.9361
SAC-mi	**0.9632**	**0.9395**	**0.9392**
random-set	0.9588	0.9316	0.9315
Salient Objects	acc	bac	f1
DQN-corr	**0.9284**	**0.6371**	**0.6596**
DDQN-corr	0.9251	0.6249	0.6458
AC-corr	0.9257	0.6285	0.6499
SAC-corr	0.9184	0.5986	0.6151
DQN-mi	0.9264	0.6312	0.6523
DDQN-mi	0.9213	0.6125	0.6306
AC-mi	0.9241	0.6193	0.6391
SAC-mi	0.9253	0.6214	0.6406
random-set	0.9244	0.6157	0.6315
Soil Moisture	r^2	mae	mse
DQN-corr	0.7983	1.2762	2.8458
DDQN-corr	0.8207	1.2265	2.5308
AC-corr	0.7987	1.3221	2.8400
SAC-corr	0.8221	1.1806	2.5098
DQN-mi	0.8011	1.3010	2.8065
DDQN-mi	0.7761	1.3618	3.1593
AC-mi	0.7850	1.2759	3.0333
SAC-mi	**0.8544**	**1.1384**	**2.0546**
random-set	0.7725	1.3535	3.2096

Table 4: Downstream supervised task modeling results, logistic and linear regression with all bands included.

dataset	acc	bac	f1
Plastic Flakes	0.9714	0.9548	0.9530
Salient Objects	0.9348	0.6790	0.7063
dataset	r^2	mae	mse
Soil Moisture	0.8436	1.1874	2.2074

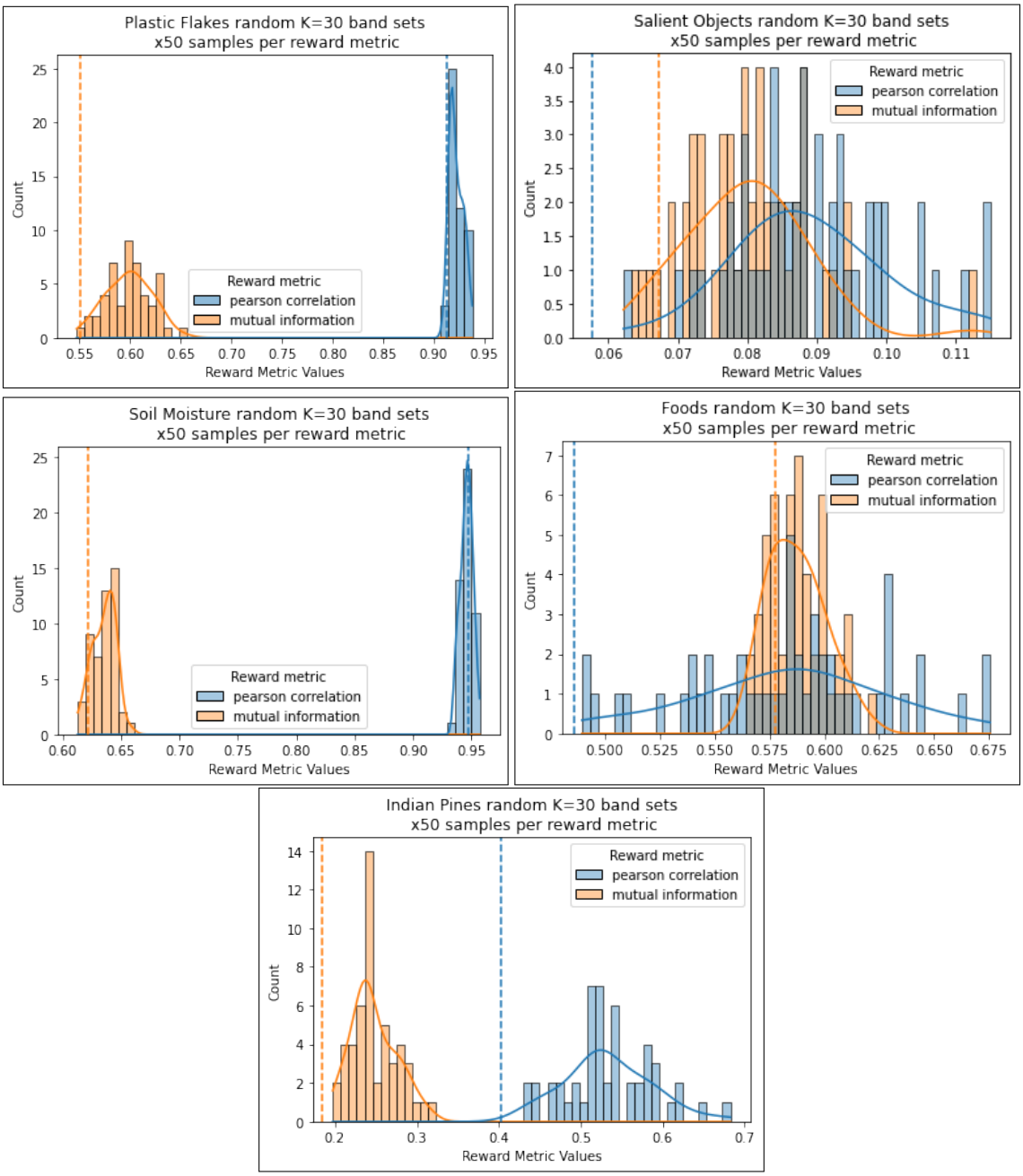

Figure 3: Random band histograms/distributions along the reward metrics (corr in blue, mi in orange). The dashed lines are the minimum reward metric observed by any RL agent after at least 750 steps (see section 4.2.3 for more details on this). Since having lower metric is better here, we can see that our best performance was in Indian Pines, Salient Objects, and Plastic Flakes. Even though some of the mutual information results from the RL agents appear to be decent in these plots (i.e., to the left of the distributions), the corresponding learning curves show that the results were likely due to chance. While the correlation results here may also be partially due to chance, their learning curves suggest that the agent actually learned to minimize this metric in some way.

is difficult. There is often very little unique "reward-to-go" difference based on the next band selection. For Q-Learning algorithms such as DQN and DDQN, we find that unless otherwise forced to make a different selection, the ArgMax of the predicted Q-values just returns the same action over and over again. We have found that the algorithms that use a network as an actor is better at handling this issue. One key reason may be that in Actor-Critic we are sampling from the action distribution being returned by the actor. In such a case, the actor may have some natural exploration and allow both the Q-network to better learn the search space, leading to greater outcomes.

Unlike correlation however, the best algorithm was image set dependent for mutual information. We believe that this is mostly driven by the fact that the Q-network had a very tough time learning the reward to go for mutual information. When reviewing the learning curves for mutual reward over time, where we see correlation generally decreasing as iterations increase, mutual information is staying static and in some instances even increasing.

One additional interesting takeaway from Table 1 is in a few instances the best correlation or best mutual information came from an agent that was trained on the other reward function. We believe that this may have something to do with the size of the dataset. For example, Indian Pines and Plastic Flakes, the two datasets with the most hyperspectral bands, both performed better with correlation. It is harder to see this distinction with the datasets with a lower number of bands.

In addition to tracking the reward function at the end of training for the agent, we also recorded the minimum reward value after iteration 750 (the point where we were confident the majority of the decisions were from the actor). Figure 3 shows the distribution of rewards from 50 random band selections. The dotted vertical line shows the minimum reward after 750 iterations from the reinforcement learning agents. Generally speaking, we find that the agent was able to find a correlation that is on the left side of the distribution. Given that we know that at iteration 750, the band selections are not random (as shown through the loss functions and repeatability of band selections), we have been able to show that the agent has likely learned a base-level understanding of the search space. However, in Food and Indian Pines dataset, we find that the agent was able to find a reward to the left of the entire distribution, showing a very good band selection.

When reviewing the downstream tasks we find that Soft Actor Critic performed the best on the Plastic Flakes dataset and the Soil Moisture dataset, while DQN performed the best on Salient Objects. We excluded the Indian Pines and Foods datasets from this table. We believe that the Indian Pines dataset labels are not correctly aligned with the pixels. For the foods dataset, all of the approaches, including the random approach, had 100% accuracy. We felt it would not be informative to include those results. In all instances, we find that the selected 30 bands perform worse than the classification task given all the bands available in the dataset. The 30 selected bands for Soil Moisture outperformed on the downstream task versus using all of the bands.

5.2 Limitations

When we first approached the hyperspectral band selection problem, we read a number of papers that touted their novel methods in this task and benchmarked results amongst each others methods. One of our key realizations after having worked with this data and conducting our own band selection is that the incremental benefit of using more complex methods to decide which bands to select is very low relative to the effort. Randomly selecting K-bands and using the set of bands with the minimum reward of choice would be an easy and effective approach to many of these problems.

In terms of other problem specific limitations, we found that many actions result in the very similar changes in selected band correlation. As such, we often found that the Q-network struggled to differentiate between different band selections. While introducing Actor-Critic type algorithms helped with this issue, it would likely be good to try and introduce other reward functions with greater discrimination between band selection choices.

The correlation and mutual information calculations were computationally intensive, especially with the larger datasets such as Plastic Flakes and Salient Objects. In those cases, we had to very sparsely sample the available data to be able to even run the model. While the introduction of a reward calculation cache helped, dealing with high compute times is an issue that is inherent with datasets of this size.

5.3 Future Directions

Further, we would have liked to try out more exploration exploitation deep reinforcement learning algorithms. Through our experiments we found out that give the models a lot of time to explore the combinations gives us better results on the trained model. This is probably because the differences in the correlation is can get really small, so the model needs more signal to be confident on its decisions.

We would have liked to benchmark the results of our model against alternative band selection methodologies, such as those that explore deep learning model parameters to search and select spectral bands. We found on experimentation

with longer runs that value seem to converge more. Due to computational and time limitations, we could not run our algorithms for longer than 2000 iterations. We also found that experimenting with exponential values of reward gave us better results. It would be interesting to explore different bases for exponent or a different function for the reward.

DRL is structured such that learning is aimed to maximize reward and not downstream accuracy. we would have liked to experiment with different reward strategies. Our entire model is based on the assumption that, interband correlation and mutual information are the best reward functions. It would be interesting to see what other reward functions work well to improve downstream accuracy.

6 Conclusion

The project aims at exploring different DRL algorithms for hyper-spectral band selection. Overall, we find that for this type of task Actor-Critic methods such as Actor-Critic and Soft Actor-Critic outperform traditional Q-Learning methods such as DQN and DDQN in terms of correlation and in many classification tasks. While we found that the agent is able to learn a valid representation of the search space, the effort and complexity of DRL (and most of the other published methods) may not be required for this problem. The vast majority of findings by us and other researchers can be comparably matched by randomly selecting bands and choosing the outcome with the best reward. However, if we wanted to keep a higher number of bands, these complex approaches may show their usefulness in navigating the larger search space.

7 Acknowledgement

8 Roles of team members

- Pratik Aher: Reinforcement Learning Experimentation, Reinforcement Learning Algorithm Implementation, Initial Data Exploration,
- Romit Barua: Reinforcement Learning Experimentation, Reinforcement Learning Algorithm Implementation, Initial Data Exploration, Writing
- Daniel Furman: Reinforcement Learning Experimentation, Initial Data Exploration, Writing, Downstream Evaluation, Dataset Curation

9 Dataset Citations

- Plastic Flakes: Open Data Commons Open Database License (ODbL) v1.0. Accessed Nov 19, 2022. link
- Salient Objects: Terms of use specified here, which were abided by. Accessed Nov 21, 2022. link
- Soil Moisture: License unknown. Accessed Nov 25, 2022. link
- Foods Dataset: Attribution-NonCommercial 4.0 International (CC BY-NC 4.0). Accessed Nov 26, 2022. link.
- Indian Pines: Open-access use as per here. Accessed Nov 19, 2022. link

References

Lichao Mou, Sudipan Saha, Yuansheng Hua, Francesca Bovolo, Lorenzo Bruzzone, and Xiao Xiang Zhu. Deep reinforcement learning for band selection in hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2022. doi:10.1109/tgrs.2021.3067096.

Ryosuke Furuta, Naoto Inoue, and Toshihiko Yamasaki. Fully convolutional network with multi-step reinforcement learning for image processing. *CoRR*, abs/1811.04323, 2018. URL `http://arxiv.org/abs/1811.04323`.

Kyle Vassilo, Cory Heatwole, Tarek Taha, and Asif Mehmood. Multi-step reinforcement learning for single image super-resolution. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2160–2168, 2020. doi:10.1109/CVPRW50498.2020.00264.

Wentian Li, Xidong Feng, Haotian An, Xiang Ng, and Yu-Jin Zhang. Mri reconstruction with interpretable pixel-wise operations using reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:792–799, 04 2020a. doi:10.1609/aaai.v34i01.5423.

Xiaolong Li, Hong Zheng, Chuanzhao Han, Haibo Wang, Kaihan Dong, Ying Jing, and Wentao Zheng. Cloud detection of superview-1 remote sensing images based on genetic reinforcement learning. *Remote Sensing*, 12(19), 2020b. ISSN 2072-4292. doi:10.3390/rs12193190. URL `https://www.mdpi.com/2072-4292/12/19/3190`.

Litu Rout, Saumyaa Shah, S. Manthira Moorthi, and Debajyoti Dhar. Monte-carlo siamese policy on actor for satellite image super resolution. *CoRR*, abs/2004.03879, 2020. URL https://arxiv.org/abs/2004.03879.

Xiangxiang Chu, Bo Zhang, Hailong Ma, Ruijun Xu, Jixiang Li, and Qingyuan Li. Fast, accurate and lightweight super-resolution with neural architecture search. *CoRR*, abs/1901.07261, 2019. URL http://arxiv.org/abs/1901.07261.

Xuan Liao, Wenhao Li, Qisen Xu, Xiangfeng Wang, Bo Jin, Xiaoyun Zhang, Ya Zhang, and Yanfeng Wang. Iteratively-refined interactive 3d medical image segmentation with multi-agent reinforcement learning. *CoRR*, abs/1911.10334, 2019. URL http://arxiv.org/abs/1911.10334.

Tuan Tran Anh, Khoa Nguyen-Tuan, Tran Minh Quan, and Won-Ki Jeong. Reinforced coloring for end-to-end instance segmentation. *CoRR*, abs/2005.07058, 2020. URL https://arxiv.org/abs/2005.07058.

Arantxa Casanova, Pedro O. Pinheiro, Negar Rostamzadeh, and Christopher J. Pal. Reinforced active learning for image segmentation. *CoRR*, abs/2002.06583, 2020. URL https://arxiv.org/abs/2002.06583.

Burak Uzkent, Christopher Yeh, and Stefano Ermon. Efficient object detection in large images using deep reinforcement learning. *CoRR*, abs/1912.03966, 2019. URL http://arxiv.org/abs/1912.03966.

Shuai Liu and Jialan Tang. Modified deep reinforcement learning with efficient convolution feature for small target detection in vhr remote sensing imagery. *ISPRS International Journal of Geo-Information*, 10(3), 2021. ISSN 2220-9964. doi:10.3390/ijgi10030170. URL https://www.mdpi.com/2220-9964/10/3/170.

Jing Yu, Deying Liang, Bo Hang, and Hongtao Gao. Aerial image dehazing using reinforcement learning. *Remote Sensing*, 14(23), 2022. ISSN 2072-4292. doi:10.3390/rs14235998. URL https://www.mdpi.com/2072-4292/14/23/5998.

Jie Feng, Di Li, Jing Gu, Xianghai Cao, Ronghua Shang, Xiangrong Zhang, and Licheng Jiao. Deep reinforcement learning for semisupervised hyperspectral band selection. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–19, 2022. doi:10.1109/TGRS.2021.3049372.

Simon J. Hook, Kevin Turpie, Sander Veraverbeke, Robert Wright, Martha Anderson, Anupma Prakash, John "Lyle" Mars, and Dale Quattrochi. NASA 2014 The Hyperspectral Infrared Imager (HyspIRI) – science impact of deploying instruments on separate platforms, 2014. URL https://hdl.handle.net/2014/45476.

Yu Wei, Xicun Zhu, Cheng Li, Xiaoyan Guo, Xinyang Yu, Chunyan Chang, and Houxing Sun. Applications of hyperspectral remote sensing in ground object identification and classification. *Advances in Remote Sensing*, 06: 201–211, 01 2017. doi:10.4236/ars.2017.63015.

Gustavo Camps-Valls, Devis Tuia, Lorenzo Bruzzone, and Jón Atli Benediktsson. Advances in hyperspectral image classification: Earth monitoring with statistical learning methods. *CoRR*, abs/1310.5107, 2013. URL http://arxiv.org/abs/1310.5107.

Bo Du, Lixiang Ru, Chen Wu, and Liangpei Zhang. Unsupervised deep slow feature analysis for change detection in multi-temporal remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 57(12):9976–9992, 2019. doi:10.1109/TGRS.2019.2930682.

Wenzhi Zhao, Lichao Mou, Jiage Chen, Yanchen Bo, and William J. Emery. Incorporating metric learning and adversarial network for seasonal invariant change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 58(4):2720–2731, 2020. doi:10.1109/TGRS.2019.2953879.

Swalpa Kumar Roy, Suvojit Manna, Tiecheng Song, and Lorenzo Bruzzone. Attention-based adaptive spectral–spatial kernel resnet for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 59(9): 7831–7843, 2021. doi:10.1109/TGRS.2020.3043267.

Weiwei Sun and Qian Du. Hyperspectral band selection: A review. *IEEE Geoscience and Remote Sensing Magazine*, 7 (2):118–139, 2019. doi:10.1109/MGRS.2019.2911100.

Mengbo You, Xiancheng Meng, Yishu Wang, Hongyuan Jin, Chunting Zhai, and Aihong Yuan. Hyperspectral band selection via band grouping and adaptive multi-graph constraint. *Remote Sensing*, 14(17), 2022. ISSN 2072-4292. doi:10.3390/rs14174379. URL https://www.mdpi.com/2072-4292/14/17/4379.

Christopher Watkins. Learning from delayed rewards. 01 1989.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. *CoRR*, abs/1509.06461, 2015. URL http://arxiv.org/abs/1509.06461.

Richard Sutton, David Mcallester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Adv. Neural Inf. Process. Syst*, 12, 02 2000. doi:10.5555/3009657.3009806.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *CoRR*, abs/1801.01290, 2018. URL `http://arxiv.org/abs/1801.01290`.

Nevrez Imamoglu, Yu Oishi, Xiaoqiang Zhang, Guanqun Ding, Yuming Fang, Toru Kouyama, and Ryosuke Nakamura. Hyperspectral image dataset for benchmarking on salient object detection. In *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–3, 2018. doi:10.1109/QoMEX.2018.8463428.

Guillem Martinez, Maya Aghaei, Martin Dijkstra, Bhalaji Nagarajan, Femke Jaarsma, Jaap van de Loosdrecht, Petia Radeva, and Klaas Dijkstra. Hyper-spectral imaging for overlapping plastic flakes segmentation, 2022. URL `https://arxiv.org/abs/2203.12350`.