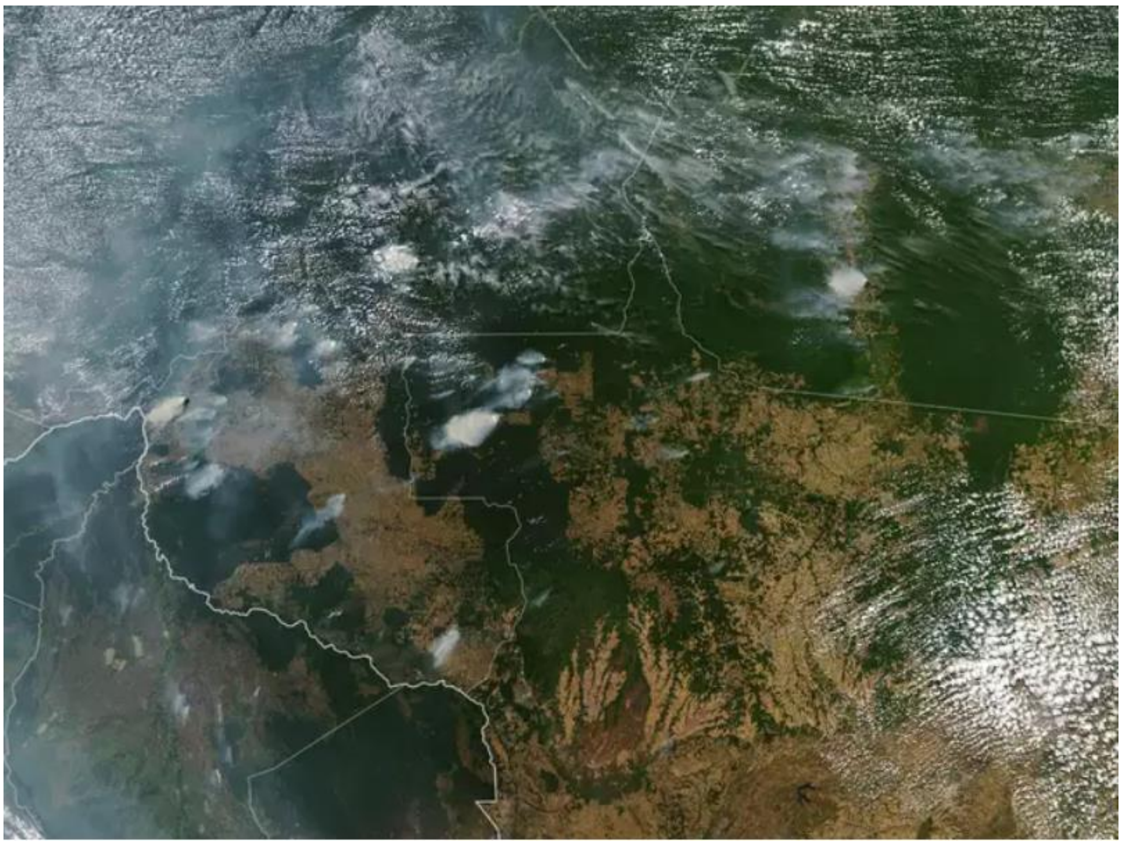
Springboard Capstone Project 2 – Milestone Report
Amazon Rainforest Image Classification



Connor McAnuff

November 14, 2019

1. Overview

1.1 Problem Statement

1.1.1 Amazon Rainforest

The Amazon rainforest covers an area of 6,000,000 km^2 across multiple countries and is located in the largest river basin in the world. It contains several million species of insect, animal, plant, and tree life. Many areas of the Amazon also contain human civilization.

Despite efforts beginning in the 1990s by the Brazilian government and international bodies to protect the rainforest, human encroachment, exploitation, deforestation and other forms of destruction continue to harm the health of the Amazon Rainforest, causing reduced biodiversity, habitat loss, climate change, desertification, and soil erosion, among other issues [1][2].

1.1.2 Satellite Imagery

Due to the immense area of the Amazon Rainforest, monitoring deforestation is difficult, if not impossible to do so from the ground. Previously, research on tracking changes in forests has been performed using satellite imagery with a resolution of 30 m/pixel (Landsat) or even 250 m/pixel (MODIS). These resolutions are too coarse to identify small-scale deforestation/degradation or identify whether the cause is human or natural. The company Planet plans to collect daily satellite imagery of the Earth's surface at a resolution of 3-5 m/pixel, which would allow for manual (and perhaps machine-learning automated) classification of Amazon Rainforest satellite images, including the location and cause of small-scale deforestation/degradation [3].

1.2 Value to client

Automated classification of Amazon Rainforest satellite images would provide great value to governments and other organizations. The scale of the area and frequency of the images means that it would be extremely cumbersome and resource intensive for images to be manually classified.

Automated classification would ideally allow for the 'human footprint' in the Amazon to be accurately and frequently mapped to identify deforestation/degradation. The human footprint map can be used by governments or organizations to take action to prevent further harm. The data would also be valuable for other reasons – for example, it could be used for infrastructure, aid, census, and resource planning.

2. Data Wrangling

2.1 Dataset Overview

The dataset to be used in for this project has been made available for a former Kaggle competition that occurred in 2017 [4]. The images were collected using Planet satellites and are given as chips (tiles). Each chip has been manually assigned a label (or multilabel) through crowd sourcing of labour. The client has stated that some mislabels are present, however this issue is common in satellite imagery classification and the proportion of mislabels is thought to be very small.

The labels stem from three categories: atmospheric conditions, common land cover/use, and rare land cover/use. The labels are unbalanced - for example there are many more images tagged with primary forest than there are images tagged with selective logging. Images commonly have more than one label.

2.2 Dataset Format

Dataset available: https://www.kaggle.com/c/planet-understanding-the-amazon-from-space/data

- 40,479 images (chips) of amazon rainforest satellite images
  - 256 x 256 pixels (947.2m x 947.2m on ground) in GeoTiff (stripped of GeoTiff metadata) and JPEG formats
  - 21.2 GB total
  - Tiff images have four bands of data: red, green, blue, near infrared (16-bit)
  - JPEG images are in the standard format of RBG arrays
  - 17 classifications (unbalanced):
    - More common:
      - Cloudy
      - Partly cloudy
      - Hazy
      - Primary rain forest
      - Water (rivers and lakes)
      - Habitation
      - Agriculture
      - Road
      - Cultivation
      - Bare ground
    - Less common:
      - Slash and burn
      - Selective logging
      - Blooming
      - Conventional mining
      - “Artisanal” mining
      - Blow down
- List of image file names and their associated labels (train_v2.csv)
  - CSV format

2.3 Data Importing

The train_v2.csv file can be read directly into a Pandas DataFrame to create a table of image names (train_0, train_1, train_2, …) and each image’s associated, manually labelled tags (i.e. labels). The tags column can have each observation split into a list of tags (strings) and then a list of all non-unique tags can be constructed to determine the frequency of each tag.

The images can be brought in for viewing using opencv methods imread and imshow.

2.4 Cleaning and Organization

There does not appear to be any missing or incorrect information in the train_v2.csv table of image names and associated tags. As all 40,000+ images cannot realistically be viewed and confirmed to be error free manually, irregularities will be searched for during exploratory data analysis.

2.5 Formatting

Dataset formatting is necessary to put the image information into the format required for neural network machine learning. This process includes the following steps:

1) Create a dictionary of tags (keys) and integers (values) that will be used to create the one hot encoding sequence.
2) Load each image one by one using keras.preprocessing.image.load_img.
3) Convert each image to a numpy array using keras.preprocessing.image.img_to_array.
4) Use the image tags and dictionary mappings to create the one hot encoder sequence for each image. The target variable for each observation is a list of 17 integers, 0 if the image does not have the tag, and 1 if the image does.
5) The numpy image array and one hot encoding sequence are appended to lists of images and targets, respectively.
6) The image and one hot encoding lists are converted to arrays X and y.
7) The dataset is saved into a compressed format (.npz) and can be loaded at any time.

3. Exploratory Data Analysis

3.1 Data Storytelling

3.1.1 Tag Occurrences

Total tag occurrences for all 40,479 images are plotted in Figure 1. The tag occurrences are unbalanced, with 37,513 and 28,431 of the images having tags primary and clear, respectively. The more important tags (those that will be used to identify deforestation/degradation) are much less common:

- Selective logging (340 images)
- Artisanal mine (339 images)
- Slash and burn (209 images)
- Blow down (101 images)

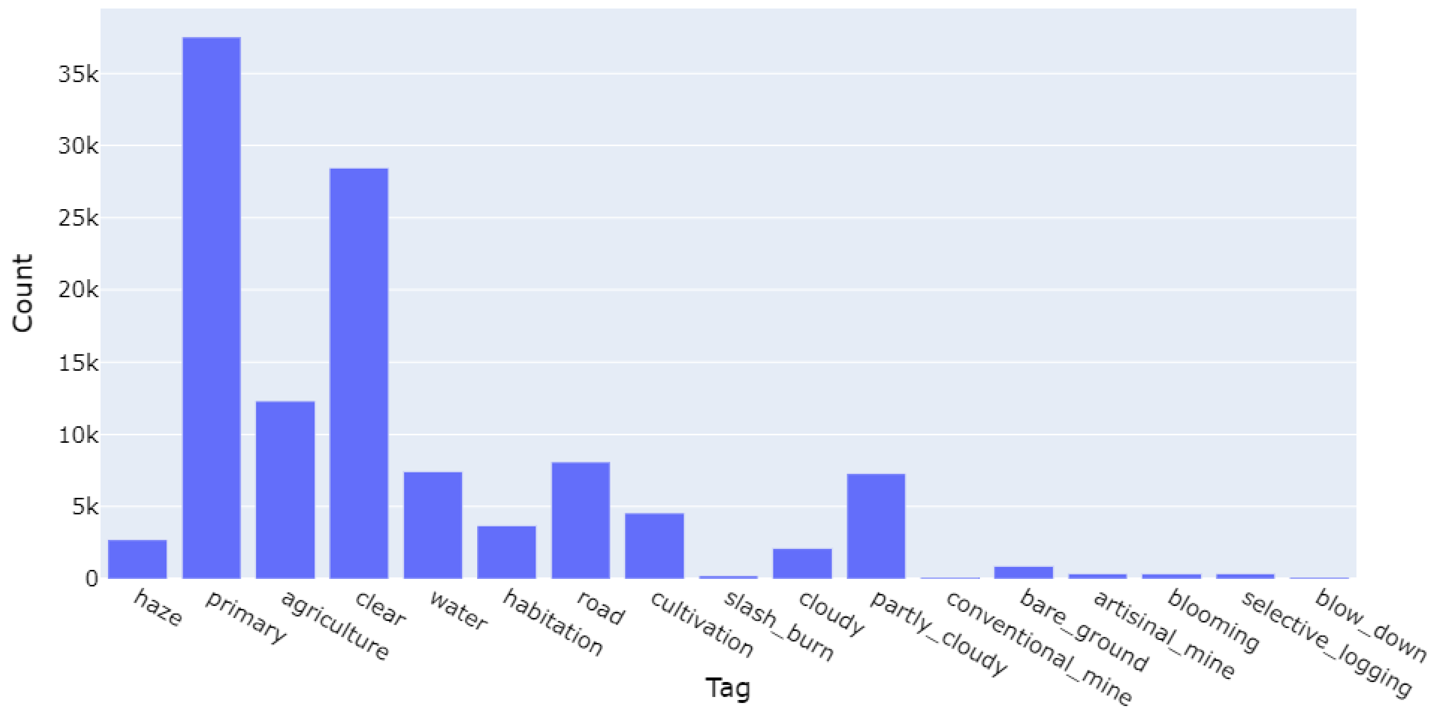


Figure 1: Total occurrences of tags for all images.

The images have an average of 2.87 tags per image. To understand which tags occur with other tags most frequently, a co-occurrence matrix is shown in Figure 2. For each tag listed on the y-axis, the co-occurrence matrix gives the proportion of images that also have each tag listed on the x-axis. For example, for all images tagged with blow_down:

- 0% of images tagged with blow_down also have tag haze.
- 100% of images tagged with blow_down also have tag primary.
- 22% of images tagged with blow_down also have tag agriculture.
- 87% of images tagged with blow_down also have tag clear.

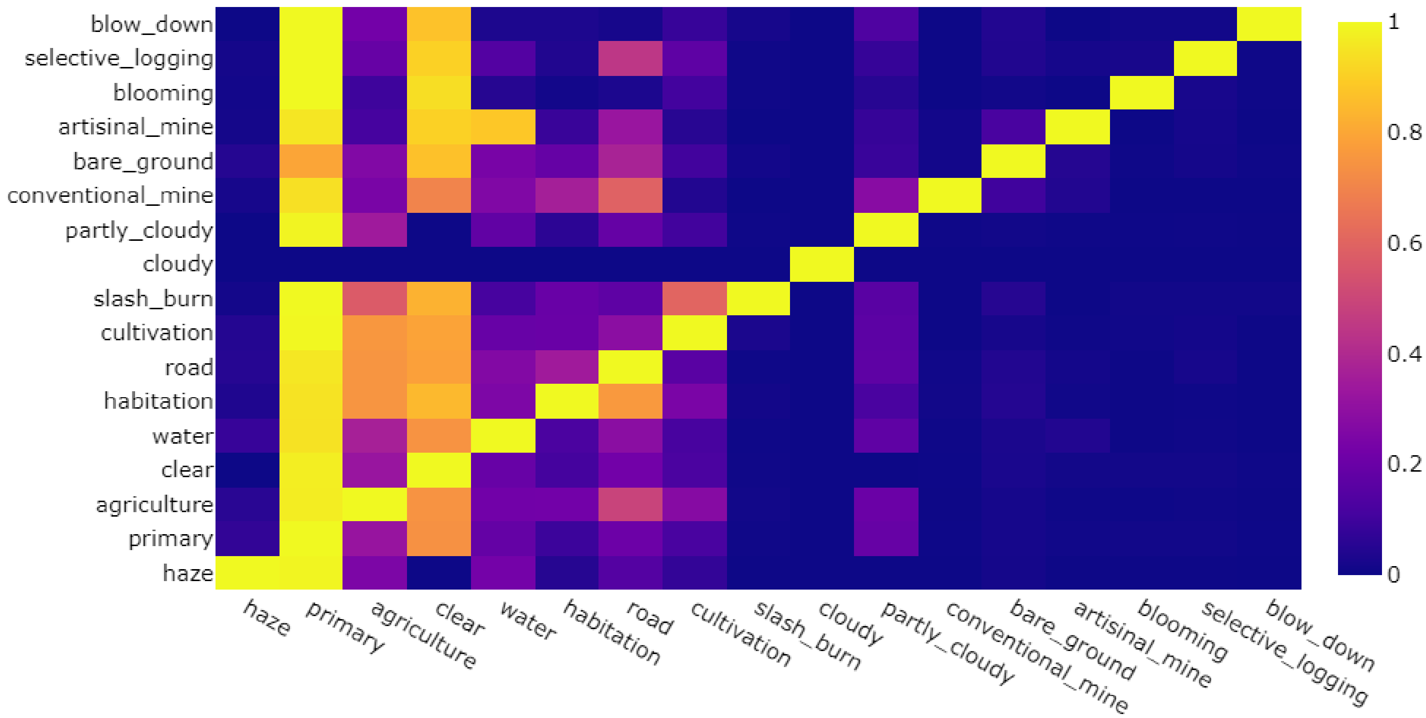


Figure 2: Co-occurrence matrix of tags. Values given are the proportion of images each tag on the y-axis that also have each tag on the x-axis.

Unsurprisingly, most tags have a high co-occurrence with primary and clear. Cloudy is the exception to this trend – images tagged with cloudy are always tagged with cloudy only. Cultivation, road, and habitation commonly occur with agriculture. Selective logging images are also tagged with road 44% of the time. Slash_burn often occurs with agriculture or cultivation.

3.1.2 Images

A sample image (train_10) is given in Figure 3. The image is 256 x 256 pixels with 3 channels for red, blue, and green intensity values. Image train_10 is tagged with primary, clear, agriculture, slash_burn, and water. Primary, water, and slash_burn are labelled in the sample image, the locations being obvious. It is not clear where the agriculture is located, however it is assumed to be the new vegetation growth nearby the slash and burn site. The tag clear applies to the entire image and is referring to the clearness of the image due to lack of clouds or haze.

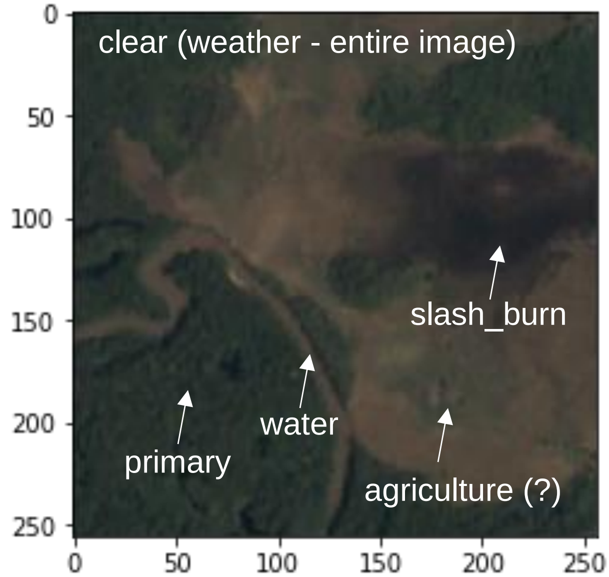


Figure 3: Sample JPEG image – train_10.

The TIFF images have 4 channels (red, blue, green, near-infrared) as opposed to the 3 channels that the JPEG images have. The additional channel, near-infrared, results in the images being ‘washed-out’ to the naked eye, as shown in Figure 4.

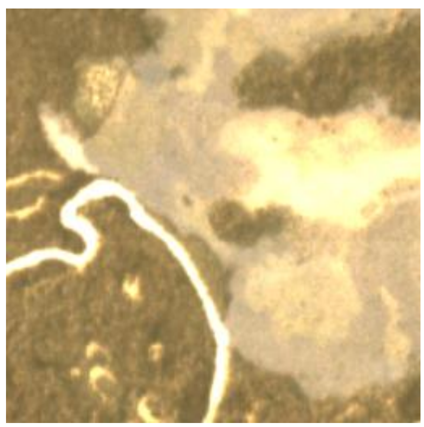


Figure 4: TIFF raw image – train_10.

The TIFF images are better viewed when each of the 4 channel intensities are shown (Figure 5). Near-infrared intensity is shown to be very low for water and dark earth, and very high for thick vegetation. For the purposes of simplicity, JPEG images will be focused on for this project.

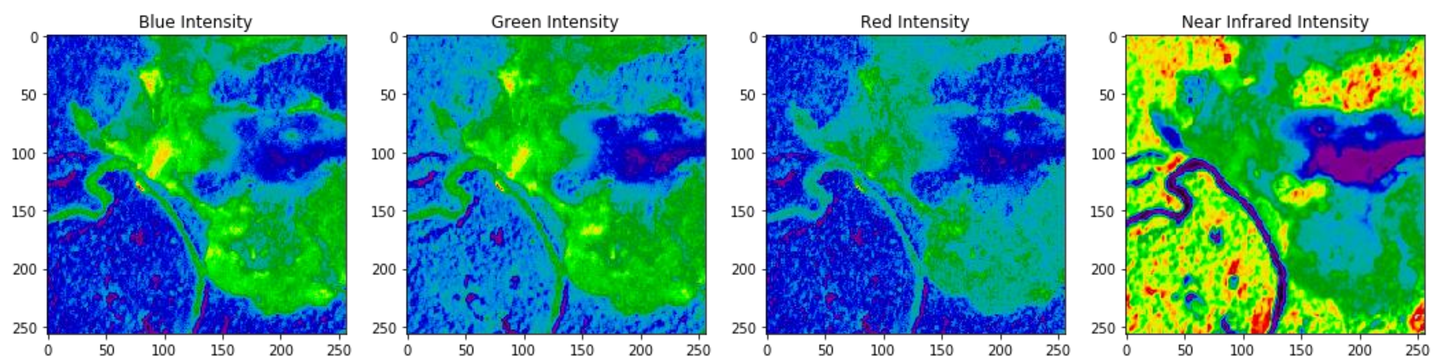


Figure 5: Sample TIFF image (4 channels) – train_10.

A further 9 image samples (in JPEG format) are shown in Figure 6. Images tagged with haze appear to have a transparent white/tan layer over the entire image. Images are tagged with water even if only a small portion of the image contains water. Images are tagged with agriculture when there are green/beige clearings in the primary forest.

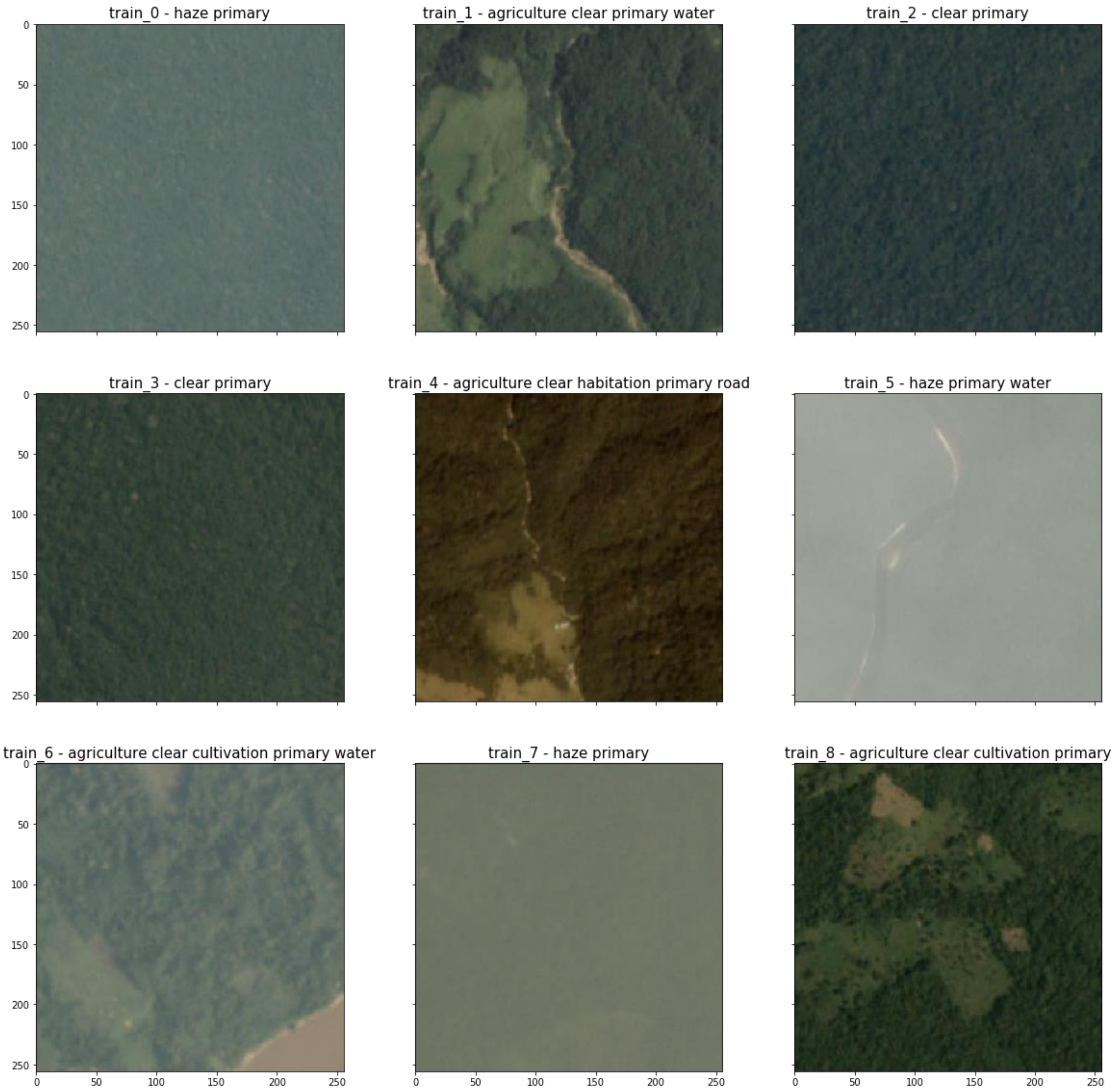


Figure 6: 9 image samples in JPEG format and their associated tags.

3.1.3 Image Histogram Equalization

Image histogram equalization is a technique used to increase contrast in images by detecting the distribution of pixel intensities (ranging 0-255) and ‘stretching’ them to better cover the pixel intensity spectrum. The technique is best understood using an example – Figure 7 shows pixel intensity histograms and images for the raw and equalized image train_8. The equalized histogram has the same shape vertically as the raw histogram, however the shape has been stretched along the x-axis to force the pixel intensities to use the entire 0-255 intensity range. The equalized image is much brighter and higher in contrast overall.

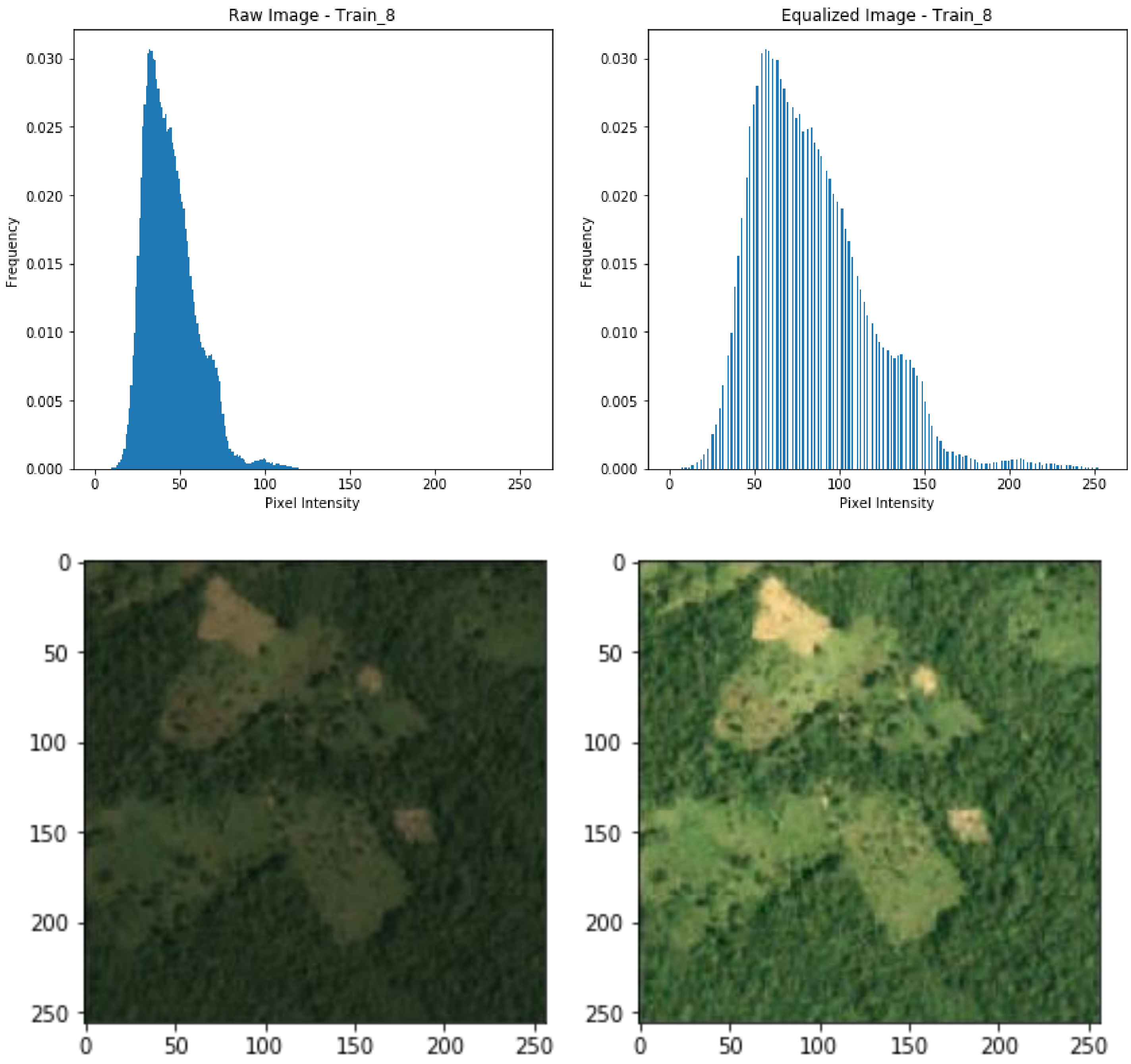


Figure 7: Raw and equalized train_8 pixel intensity histograms and images.

Image equalization can be used as a method of image augmentation when training image classification models. The modified images may allow for the machine learning model to better identify features and therefore provide more accurate classification results.

3.1.4 Single Image Means and Dominant Colours

Single images can be simplified by calculating their mean colour. The mean colour is calculated using the following process:

1) Take the mean value of the numpy array row-wise producing the mean R, G, and B values for each column.
2) Again take the mean value of the numpy array row-wise producing the mean R, G, and B values for all pixels.
3) Round the float means and convert to integers.
4) Multiply the R, G, B mean array by an array of ones in the shape of the original image. Each pixel of the final mean image has the mean pixel values.

A mean image sample is given in Figure 8. The mean image appears to be a mix of brown and green, as expected. It is not very exciting to look at and doesn't say much about the image itself, as the mean image colour does not necessarily appear frequently in the raw image. It would be more informative to explore the dominant colours of an image.

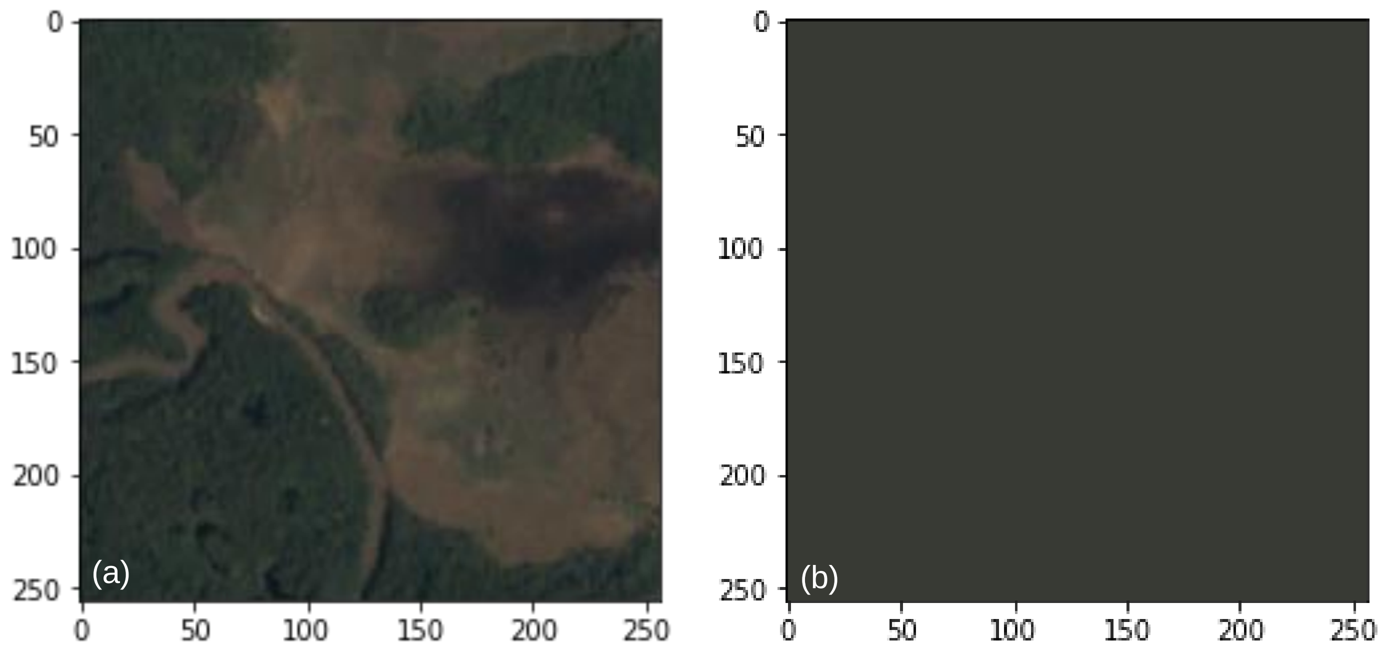


Figure 8: train_10 raw (a) and mean image (b).

Obtaining the dominant colours in an image can be performed by k-means clustering. The process is as follows:

1) Reshape the image array from (256, 256, 3) to (65536, 3).
2) Set the iteration criteria.
3) Initiate flags to be random centers.
4) Perform k-means clustering with the selected number of clusters/colours to obtain labels and centers (RGB colours).
5) Calculate the number of pixels in each label cluster.

6) Use numpy.argsort to get the sorting indexes of the number of pixels array and reverse the list.
7) Create a cumulative frequency array holding the proportion of pixels with each label.
8) Calculate the location of the separation between dominant colour rows according to the image shape.
9) Create the dominant colour image by filling the rows with the dominant colours according to their proportions.

The dominant colours of image train_10 according to k-means clustering are shown in Figure 9. Depending on whether 3, 4, or 5 clusters are implemented, the resulting dominant colour image shows various shades of green and brown, as can be seen in the raw image.

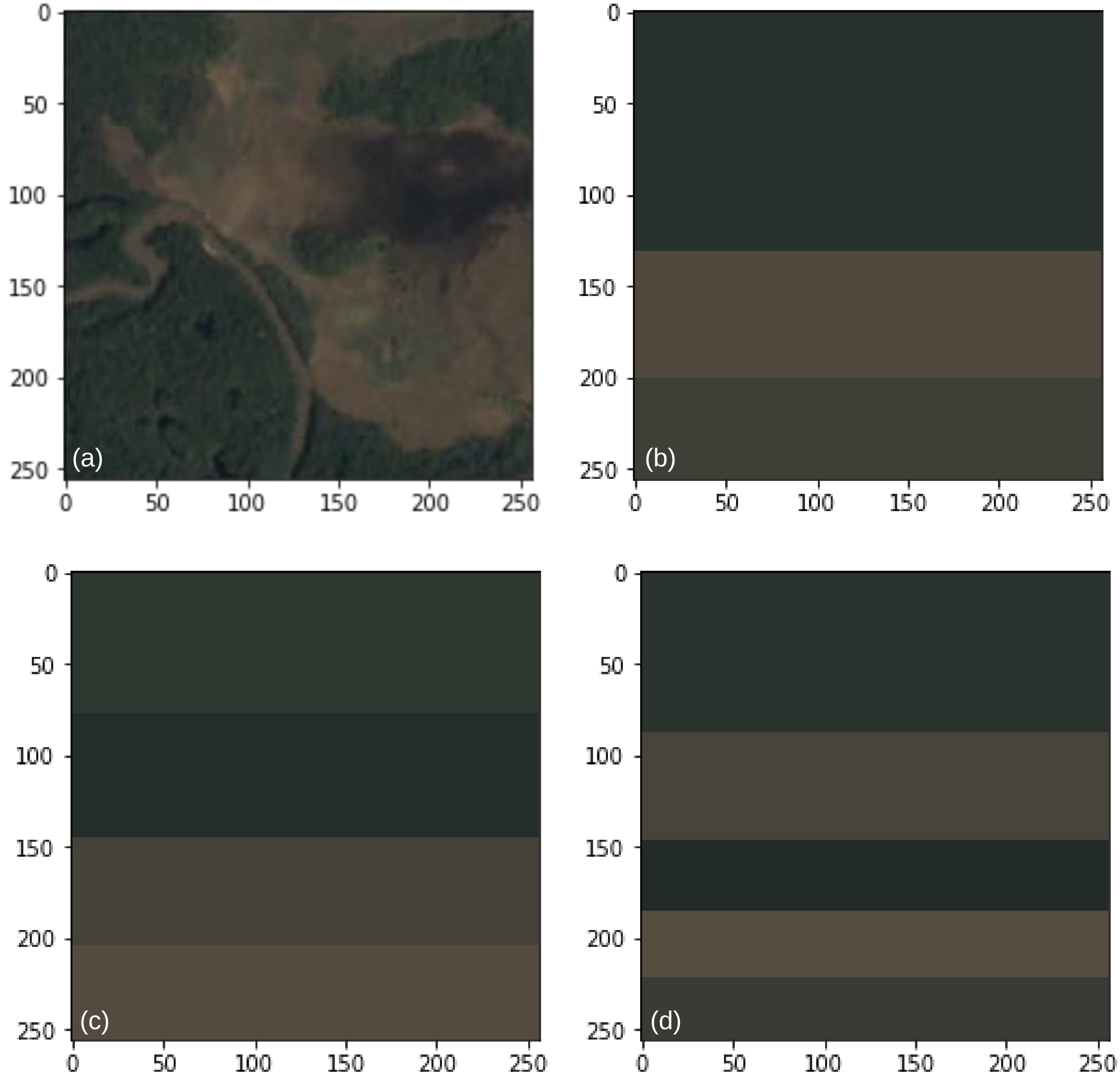


Figure 9: Dominant colours of image train_10 found using k-means clustering: (a) Raw image, dominant colours using (b) n_clusters=3, (c) n_clusters=4, (d) n_clusters=5.

Extracting the dominant colours removes information from the image in that the features that would be searched for by a deep learning algorithm are removed. The dominant colour images are a function of the colours in the image and their relative frequency. Images tagged with road will have varying amounts of ‘road pixels’ in the image. This process is unlikely to be useful for machine learning models but is an interesting exercise none-the-less.

Lastly, a tool to replace each pixel by the dominant color cluster that it belongs to has been created. Figure 10 shows images where the raw image pixels have been replaced by their associated dominant colour cluster using a varying number of clusters. As the number of clusters increases, so does the detail in the image. At 50 clusters (50 dominant colours used to re-create the image), the image is essentially indistinguishable from the raw image. This method is a method of colour quantization, wherein the number of distinct colours used in an image is reduced to compress the image file size.

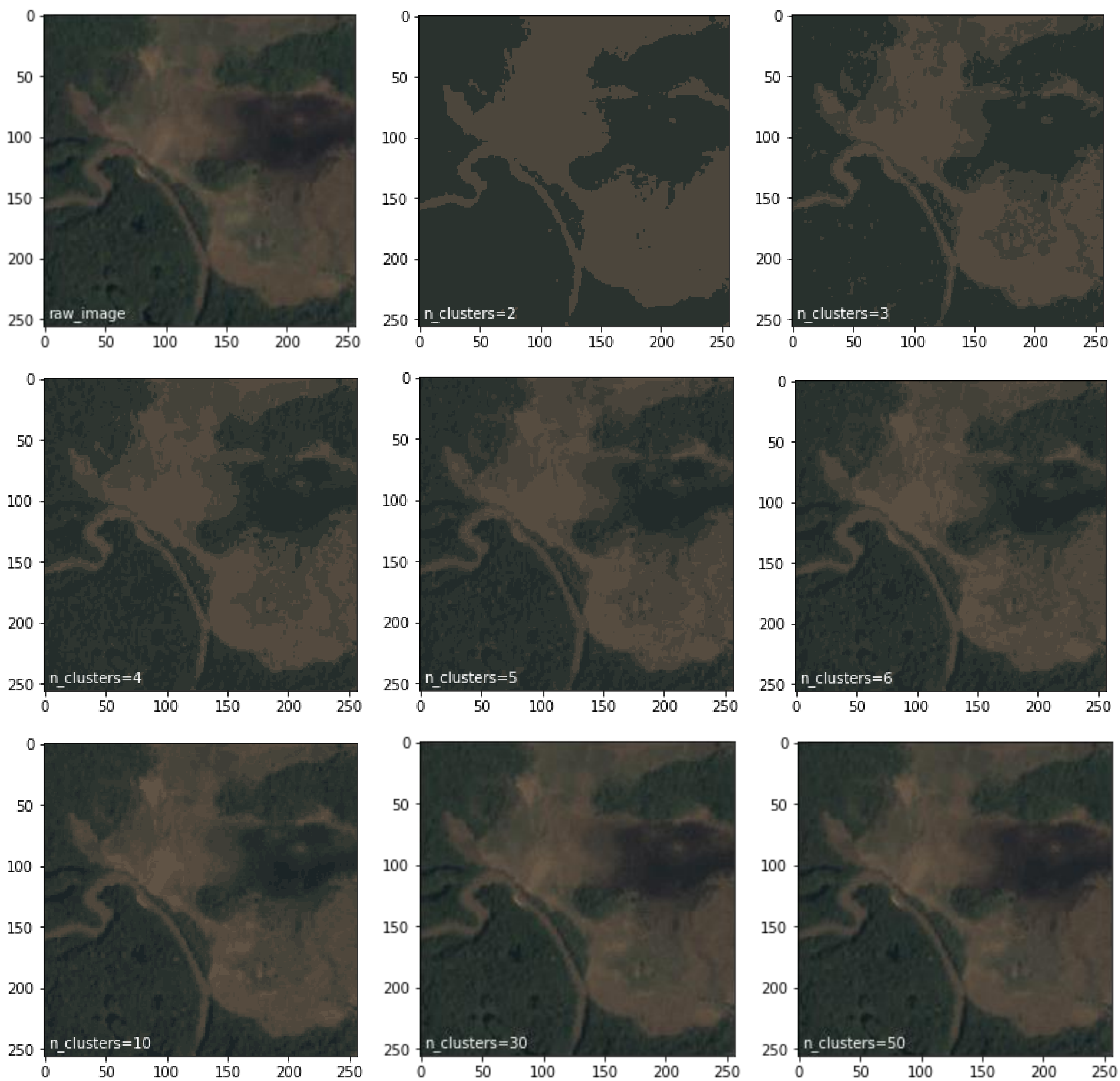


Figure 10: Image pixels represented by their associated dominant colour cluster.

3.1.5 Image Set Means

Sets of images can be combined into a single mean image by the following process:

1) Initialize the mean image as a numpy array of all zeros with the same shape as the raw images.
2) For each image in the set, add the image to the mean image.
3) Divide the mean image by the number of images in the set.
4) Round the float values and convert to integers.

The mean image for all images combined is shown in Figure 11. It is a shade of green which makes sense as most of the images are tagged with primary and thus have trees in the image.

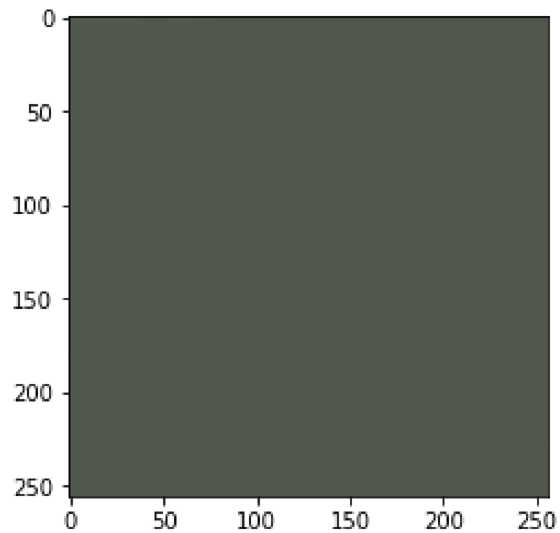


Figure 11: Mean image for all images combined.

A set of image lists can be constructed for which each list contains the list of images tagged with a given tag. The mean image for all images with that tag can then be calculated.

The mean images for each tag are given in Figure 12. The mean images by tag can give an idea of the colour scheme and prevalence of tagged information in the image sets. Of the more rare tags, selective_logging, blooming, slash_burn, and blow_down appear to be the most similar to each other and the primary mean image. Conventional_mine and artisinal_mine are very different from primary, and different than each other. Overall, there is at the least slight variability between tag mean images, indicating images with different tags do indeed have varying colours.

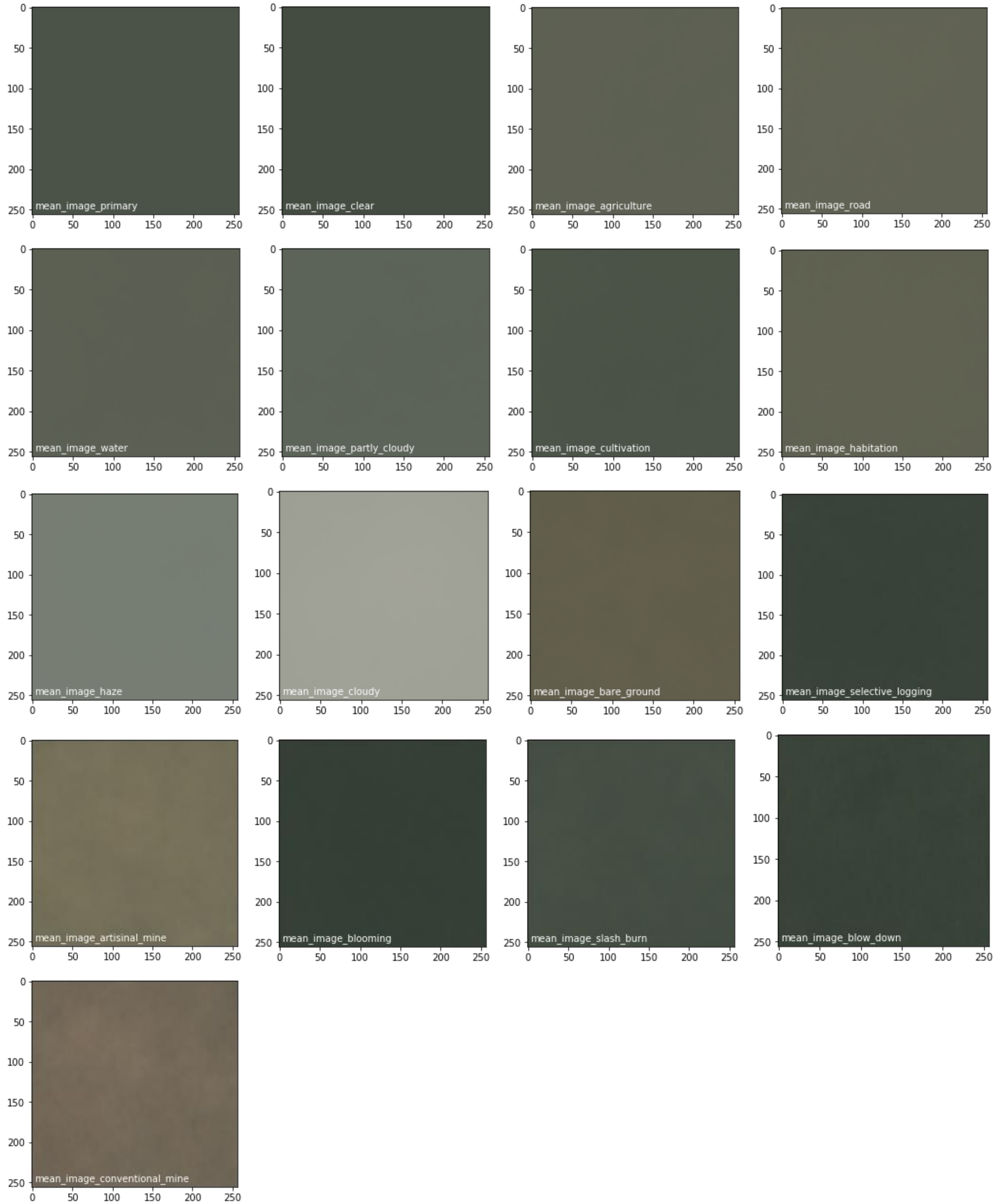


Figure 12: Mean images for sets of images with each tag.

3.1.6 Image Clustering

The images can be clustered to determine similarity using the TSNE method. For this clustering task, images tagged with only one of the rare image tags are included:

- Bare_ground
- Selective_logging
- Artisinal_mine
- Blooming
- Slash_burn
- Blow_down
- conventional_mine

The clustered images are plotted on using the two components of the TSNE clustering as the x- and y-axes (Figure 13). This plot shows that there are somewhat well-defined clusters, and several outliers. The size of the images obscures the location of many of the points – the images would be better plotted as points.

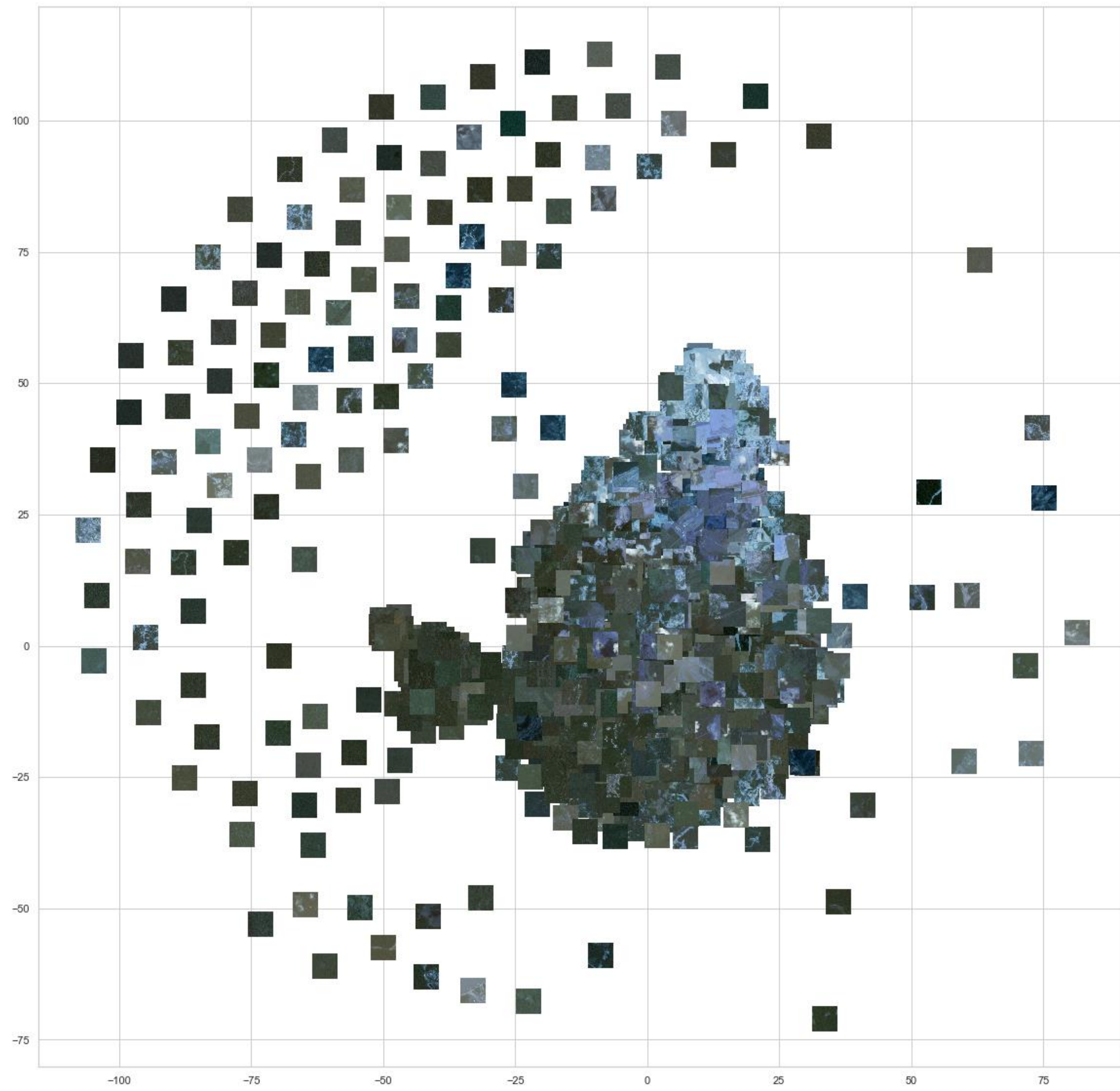


Figure 13: TSNE clustering of rare tag images.

In addition to plotting the images as points, the points can be clustered according to their rare tag to compare the results of the clustering to the image tags (Figure 14). Figure 15 shows a magnified section of the original plot.

There are multiple clusters that correspond well to rare image tags. The following observations can be made:

- Images with tag bare_ground are clustered in 3 locations and are spread around outside of all clusters.
- Images with tag artisanal_mine are partially clustered with bare_ground images. Artisanal mining would clear the land vegetation, leaving the ground bare.
- Images with tag blow_down are loosely clustered (at (-40, -5)) and are also spread around.
- Images with tag conventional_mine are not well clustered.
- Images with tag blooming are clustered in 2 locations and are also spread around.
- Images with tag slash_burn are spread around.
- Images with tag selective_logging are well but loosely clustered in the largest cluster.

Overall, there are many images that do not cluster well with others with this method. Clustering will not be able to be used to predict the labels of images accurately enough - a neural network will need to be used to identify features in the images on a smaller scale.

Figure 14: TSNE clusters with points coloured by rare tag.

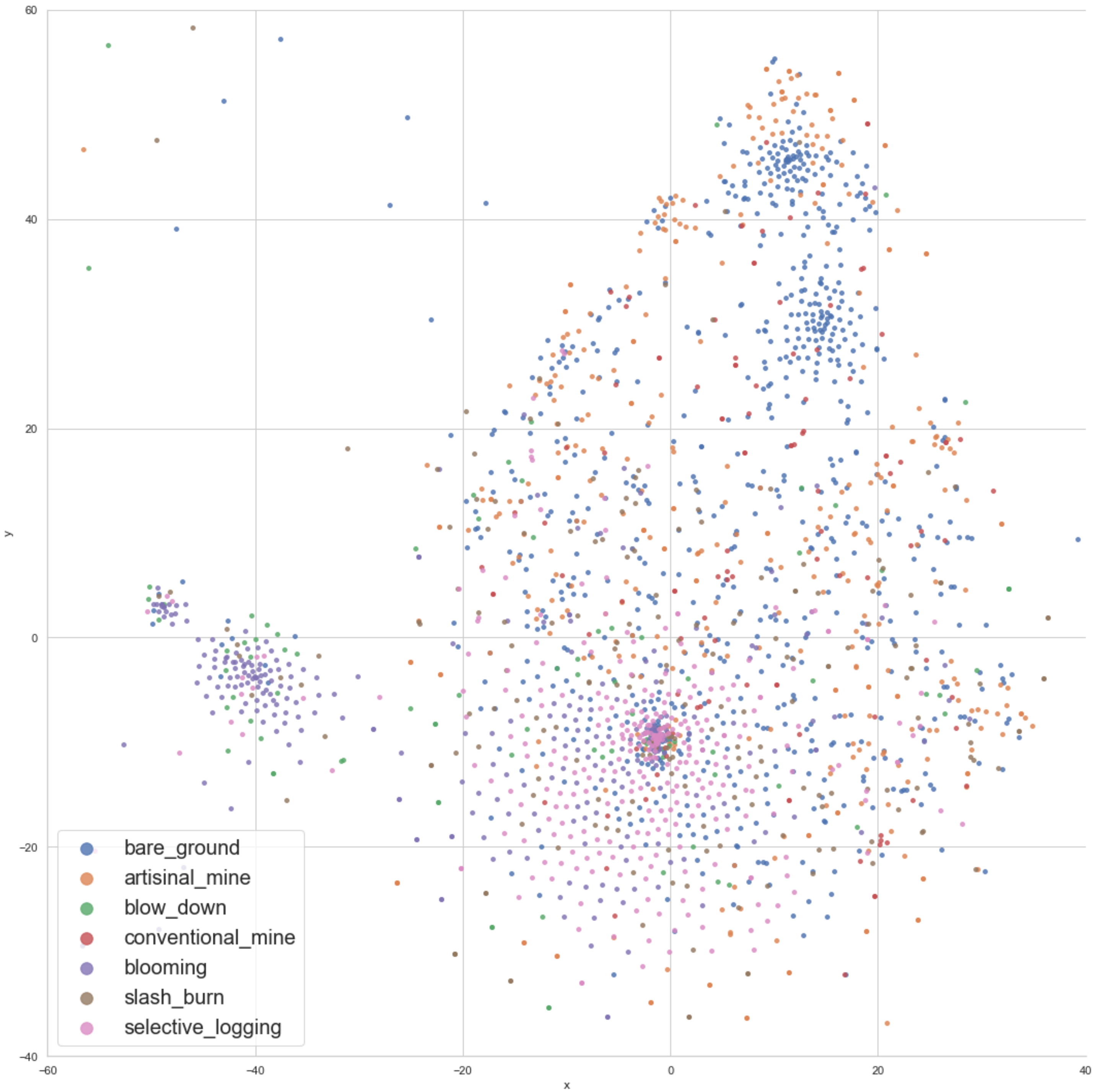


Figure 15: TSNE clusters with points coloured by rare tag (magnified).

4. References

[1] Encyclopaedia Britannica, "Amazon Rainforest," 27 08 2019. [Online]. Available: https://www.britannica.com/place/Amazon-Rainforest. [Accessed 2019].

[2] Pachamama Alliance, "Effects of Deforestation," 2019. [Online]. Available: https://www.pachamama.org/effects-of-deforestation. [Accessed 2019].

[3] Planet, "Understanding the Amazon from Space," 2017. [Online]. Available: https://www.kaggle.com/c/planet-understanding-the-amazon-from-space/overview/description. [Accessed 2019].

[4] Planet, "Understanding the Amazon from Space - Data," 2017. [Online]. Available: https://www.kaggle.com/c/planet-understanding-the-amazon-from-space/data. [Accessed 2019].