Exploring Mediatoil Imagery: A Content-Based Approach

by

Sahil Saroop

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the MCS degree in
Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

The future of Alberta's bitumen sands, also known as "oil sands" or "tar sands," and their place in Canada's energy future has become a topic of much public debate. Within this debate, the print, television, and social media campaigns of those who both support and oppose developing the oil sands are particularly visible. As such, campaigns around the oil sands may be seen as influencing audience perceptions of the benefits and drawbacks of oil sands production. There is consequently a need to study the media materials of various tar sands stakeholders and explore how they differ. In this setting, it is essential to gather documents and identify content within images, which requires the use of an image retrieval technique such as a content-based image retrieval (CBIR) system. In a CBIR system, images are represented by low-level features (i.e. specific structures in the image such as points, edges, or objects), which are used to distinguish pictures from one another.

The oil sands domain has to date not been mapped using CBIR systems. The research thus focuses on creating an image retrieval system, namely Mediatoil-IR, for exploring documents related to the oil sands. Our aim is to evaluate various low-level representations of the images within this context. To this end, our experimental framework employs LAB color histogram (LAB) and speeded up robust features (SURF) in order to typify the imagery. We further use machine learning techniques to improve the quality of retrieval (in terms of both accuracy and speed). To achieve this aim, the extracted features from each image are encoded in the form of vectors and used as a training set for learning classification models to organize pictures into different categories. Different algorithms were considered such as Linear SVM, Quadratic SVM, Weighted KNN, Decision Trees, Bagging, and Boosting on trees. It was shown that Quadratic SVM algorithm trained on SURF features is a good approach for building CBIR, and is used in building Mediatoil-IR.

Finally, with the help of created CBIR, we were able to extract the similar documents and explore the different types of imagery used by different stakeholders. Our experimental evaluation shows that our Mediatoil-IR system is able to accurately explore the imagery used by different stakeholders.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Dr. Herna Viktor, for her excellent and valuable supervision, support, and guidance throughout my graduate studies as well as specifically during the completion of this thesis. Our inspiring discussions and her insightful feedback have made positive contributions to every chapter of this thesis.

I would also like to express my profound thanks to my co-supervisor, Dr. Patrick McCurdy; his expertise and constructive comments and our fruitful discussions were of great assistance in the preparation of this thesis. I appreciate his detailed reviews, feedback, and suggestions on every chapter of this thesis.

I would like to thank my colleagues Prof. Dinesh Kumar, and Prof. Rajeev Vashisht for their constructive discussions and friendship.

A special thanks also goes to my parents for the support and trust they have always shown to me and for inspiring me to seek higher education.

Last but not least, I am grateful to my friend Sheenam Sharma for her continued support and encouragement.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The sizes of digital image repositories continue to expand with the development of the Internet, reduction in data-storage costs, and improvements in technologies for image-capturing devices (e.g., digital cameras and image scanners, amongst others). To deal with the high volume of pictures, users require dynamic image searching and retrieval tools. Three fundamental frameworks for retrieving images exist: one technique uses textual metadata, a second applies content-based image retrieval (CBIR), and a third combines the first two techniques. The first method requires assigning keywords to each picture, which is laborious, human intensive, and less appropriate, as different individuals may have different interpretations of the same thing [43]. Instead of using keywords, the CBIR technique uses the query image as an input and fundamentally matches low-level features. Humans usually describe the objects in photographs at a very high level, for example as a landscape or a plane; in contrast, computers describe them using low-level features and identify edges, corners, and objects, amongst other items. In the last few decades, a number of commercial products and experimental prototype systems have been developed, including QBIC [23], SWIM [113], Photobook [72], Virage [26], VisualSEEK [85], Netra [47], MARS [61], and SIMPLIcity [103].

Machine learning has long been used in conjunction with image retrieval systems. It has aided the quality of retrieval, in terms of both accuracy and precision, by either grouping images into different categories or classifying a given image based on its unique attributes and predicting its relevant class [43, 54, 55, 87, 97, 114]. Furthermore, CBIR combined

with machine learning have been used in different applications to query and analyze images [43]. The application domains of these systems include remote sensing, history, fashion, security, crime prevention, biodiversity information systems, publishing, art collections, retail catalogues, medical information retrieval, face finding, and architectural and engineering design [43]. Despite the many CBIR application areas, one field that is still unexplored is the contested media framing of Canada's oil sands. Our research develops a system that targets this application area.

In this study, we use supervised machine learning to label images in different categories. Our intention is to explore how pictures related to Canada's oil sands, as obtained from various stakeholders, vary in content. Moreover, we use existing CBIR techniques combined with machine learning to implement an image retrieval system named "Mediatoil-IR." As a CBIR system works on low-level features, we experiment with different low-level representations of the pictures. We also evaluate the ability of training models prepared using machine learning classifiers to assign pictures to six categories. The purpose is to choose the best of both worlds and create an efficient image retrieval system that retrieves similar images with greater precision.

The motivation, goals, and organization of our thesis are presented below.

1.1 Motivation

Campaigns and debates over Canada's energy future with its oil sands have become a flashpoint for political action. Stakeholders driving the discussions may be divided into three broad groups, namely civil society, industry, and government [59]. By publishing reports, creating still advertisements, and posting videos, oil sands stakeholders seek to influence how the public views oil sands development and ultimately the fate of this natural resource [59].

Opponents and critics have identified advantages and drawbacks to continuing to develop the oil sands [10, 19, 67, 71]. On the positive side, supporters see the oil sands as a resource that can meet the growing global demand for energy [10, 67]. The machinery and other goods required in oil sands production are produced in Central and Eastern

Canada, which provides economic benefits to various sectors [10]. Oil sands development also provides economic benefits to Canada by creating jobs and generating taxable revenue [10]. According to a report by the Canadian Association of Petroleum Producers (CAPP), "*Direct employment in Canada as a result of oil sands investment is expected to grow from 151,000 jobs in 2015 to over 350,000 jobs in 2035*" [10]. In the same report it was further found that "*Over the next 20 years, the oil sands industry is expected to pay $1.2 trillion in provincial and federal taxes – including royalties. These revenues contribute to government spending on infrastructure, social services, and other important programs. A healthy oil sands industry results in higher revenues for governments*" [10].

Despite their numerous benefits, the tar sands also pose environmental and social challenges [19, 67, 71]. The synthetic crude derived from bitumen is more carbon-intensive than that obtained utilizing conventional methods [19, 71]. Moreover, processing bitumen requires using massive amounts of water, which may come from fresh water sources. According to the authors in [71], "*Oilsands companies are not required to stop withdrawing water from the Athabasca River, even if river flows are so low that fisheries and habitat are at risk.*" Furthermore, the tailing ponds contain acutely toxic compounds that can be harmful to aquatic organisms. Authors in [71] also stated that "*Oilsands development threatens to harm millions of birds through habitat fragmentation and destruction.*" Land disturbance and use are additional concerns [19]. Restoring boreal forests and wetlands to their native state is a major challenge [71]. The land used in mining may be restored in 30-40 years, but practices that claim to restore land in 8-10 years are being implemented [67].

Given an assumption that media can be a quite powerful tool for impacting audience opinions on the oil sands development, different visuals used by the various stakeholders need to be explored. This exploration requires a CBIR system for image retrieval. Despite various application areas of CBIR systems, the domain of oil sands campaigns and their images is still unexplored. While multiple accounts exist, reviewing the tar sand's history reveals that little work has been done on contested media containing bituminous sands [59]. To bridge this gap, we are interested in discovering how the struggle over Alberta's oil sands has played out in the reports, publications, and campaigns of different tar sands

stakeholders. We pay particular attention to the graphics, images, videos, and text produced by various organizations, as they are vital resources in any media and thus critical for understanding the context, evolution, and essential characteristics of the oil sands [59].

The next section explains some of the goals we identified to achieve our objective of exploring tar sands imagery.

1.2 Thesis Objective

Our research has two primary goals. First, we aim to study and use different image representation techniques to explore the contested media related to the Canada's tar sands and obtain deep insights into the type of imagery utilized by various stakeholders. Secondly, we aim to create a CBIR system for the oil sands images. In addition to helping our research, this CBIR will also be made publicly available to assist researchers in other relevant fields. They in turn can then use it to retrieve similar images and details from the database, which may be beneficial to their own work.

To achieve our objectives, we identified six sub-goals (with the first two being preliminaries for the actual research). The first was to obtain or create an open data repository. In various research areas, "*the availability of open datasets is considered as key for research and application purposes*" [37]. As no existing database was available, it became essential in our work to create a database to persistently store different types of media (e.g., images, videos, and reports) produced by various organizations in their ongoing struggle to influence public perceptions of the oil sands. It then became apparent that an appropriate medium for fetching and visualizing the content within the dataset was much needed. Our second aim was thus to create a website that allows an appropriate audience and researchers to retrieve the documents related to tar sands.

Once the research began, the third purpose was to train machine learning-based classifiers to categorize images based on their content into six categories, namely open-pit, machines, landscape, people, protest, and graphics. The fourth aim was to study and report the similarities or differences within the media produced by various stakeholders over the years. Our fifth sub-goal was to create an efficient CBIR system, known as Mediatoil-IR,

as none of the currently available CBIR systems deal with oil sands. Our method divides images into six identified categories, creating a separate search dictionary for each class and seeking an appropriate dictionary for image matching and retrieval. Categorizing an image requires learning classification models that use different supervised machine learning-based classifiers. With reference to the "no free lunch" theorem [9], no single classifier is best. This theorem states that "*There is no one model that works best for every problem, so it is a common practice in machine learning to try alternative models and find one that works best for the given problem*" [9]. Our final aim was thus to evaluate the performance of different machine learning classifiers vis-à-vis the oil sands imagery.

1.3 Thesis Organization

The remainder of this thesis is organized as follows. In Chapter 2, we specify the background work, including the general framework of CBIR systems and the previously introduced low-level image features employed in image processing. We also explain the different detectors and descriptors utilized in image processing and detail the use of supervised machine learning for picture classification. Chapter 3 presents a case study of the Mediatoil project and introduces the Mediatoil database that was formed to create a repository for image retrieval. It also includes a discussion of the Mediatoil website, which we built as a part of our thesis to visualize oil sands pictures from different perspectives. Chapter 4 involves the analysis of the various types of images used by various stakeholders to impact human perceptions of the cost and benefits of the Canada's tar sands. In Chapter 5, we explain the Mediatoil-IR, i.e. the methodology we employed to accomplish image retrieval in the Mediatoil project. Chapter 6 presents the experiments and evaluations of the different techniques and algorithms that we used to construct the Mediatoil-IR. Finally, Chapter 7 discusses the study's conclusions and identifies areas for future work.

Chapter 2

Background Work

To achieve our goal of exploring the images provided by different oil sands stakeholders in Canada, we need to understand each picture's inner content. This implies that the images need labels that represent their content. Labeling may be done manually or by choosing other alternatives, such as training a machine learning-based classifier to predict a picture's class. As we also aim to build a CBIR system that can retrieve imagery related to the tar sands, we need to understand two things: how a CBIR system works and how machine learning can improve the accuracy and speed of CBIR.

This chapter presents an overview of various terminologies and techniques that are used in the field of image retrieval. It also discusses various low-level image features, such as color, texture, shape, and spatial location. Furthermore, it explains different detectors and descriptors for building an efficient image retrieval system. The last section addresses the use of machine learning in image categorization. We begin below by discussing the framework of image retrieval systems.

2.1 The General Architecture of CBIR Systems

Content-based image retrieval refers to the retrieval of images based on the image features extracted from a query image itself, without inspecting the keywords or annotations related to it [84]. As shown in Figure 1 [84], to retrieve images, a CBIR system first needs to store the features of all of the given images. The initial step in training is thus to collect pictures. This image set is then fed to a feature extractor, which extracts the global or local features

from the image (depending on the application needs) and encodes them into numeric values called feature vectors. Thereafter, the image indexer indexes all of the features for fast retrieval. A complete discussion of the method we employ to create search dictionaries is included in Section 6.1.4.

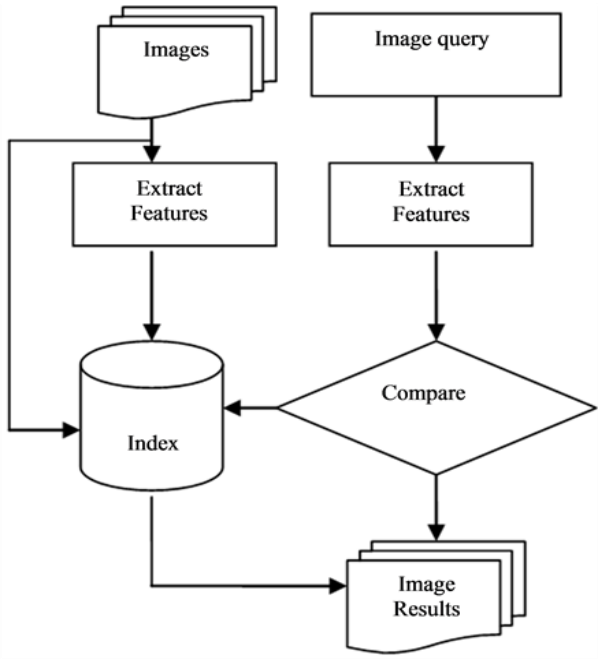


Figure 1. A framework for CBIR [84]

The extraction process takes the query image as input, extracts features from it, matches elements with the dictionary created in the training phase, and retrieves the most relevant results based on similarity values (as calculated using different distance calculation matrices) in the ranked order [84].

The next section presents a detailed discussion of different low-level features.

2.2 Low-Level Image Features

Intuitively, the way in which humans perceive images differs from the way in which computers handle them. People usually describe pictures at a very high-level, such as they contain flowers, a plane, or a book; in contrast, machines specify them at a low-level, using keypoints that identify inter alia edges, corners, and objects, amongst others. The following discussion focuses on various low-level features used in image processing.

2.2.1 Color Features

A color feature is prevalent and commonly used in image retrieval. As suggested by authors in [54], "*when finding an image collection made up of beaches, cities or highways, it is desirable to use a global image feature such as a color histogram that captures the color component of the entire scene.*" Some common color spaces that are closer to the way in which humans perceive colors are discussed next.

In the red-green-blue (RGB) color space, a pixel is represented by specifying its red, green, and blue components [104]. The RGB color model was commonly used in the past due to its easy-to-understand encoding, and computer monitors still use it to display colors [104]. The drawback of the RGB model is that it represents a non-linear color space. In such a color space, doubling the number that represents a color channel's intensity does not necessarily double the effect in visual perception. This effect is expected as it follows some non-linear curves, which makes it a less suitable choice in color manipulation software [80]. Another color model cyan-magenta-yellow-key (CMYK), which contains cyan, magenta, yellow, and key (black) as its primary color components, is used in color printing [106]. The drawback is that it is also non-linear, similar to the RGB space.

The LAB color space is a model with a wider range than RBG or CMYK [105]. In the LAB color space, 'L' stands for lightness, while 'a' and 'b' are the color-opponent dimensions. The most important attribute of LAB is that it is device independent, i.e. it defines colors independently of their origin or the device on which they are displayed [105]. The other important property of LAB is that it is perceived as a linear color space. Such a color space handles the relationship between the numbers they store and the intensities they represent in a linear manner, i.e. doubling the number results in doubling the intensity [80]. The property of being linear makes the LAB color space well suited for color manipulation software [80]. As reported in [80], the only drawback is that one needs to have an understanding of the color temperature scale to create colors manually, although this ability is built into color manipulation algorithms.

Hue, saturation, and value (i.e. brightness) constitute a cylindrical coordinate representation of a point in an RGB model and are simple transformations of device-

dependent RGB models. Each unique RGB device thus has a unique hue-saturation-value (HSV) space [104]. On the positive side, HSV is easy to understand; on the downside, however, it is again non-linear, similar to other color spaces (apart from LAB). To learn more about different color spaces, including RGB, LAB, HSV, CMYK, and hue-min-max-difference (HMMD), readers are directed to [44, 49, 62, 82, 89].

Once the color space has been selected, the next step is to determine which features to use and how to encode an image within a feature set. Several color description techniques have been proposed in the literature. These methods are divided into two categories, depending on whether they include information related to spatial distribution [17].

It is worth noting that a color histogram [43, 54, 92], which is the most widely used descriptor in image retrieval, does not consider spatial color distribution. The color histogram extraction algorithm includes three steps: partitioning the color space into cells, associating each cell to a histogram bin, and counting each cell's image pixels and writing this number in the associated histogram bin. This descriptor is translation and rotation invariant. The similarity between two color histograms can be computed in various ways, such as by using the L1 (Manhattan distance), L2 (Euclidean distance), weighted Euclidean distances or by calculating their intersection [92].

Similar to the color histogram, another example of a descriptor that does not consider a color's spatial distribution is color moments [50]. The mean (first order), variance (second), and skewness (third) are most commonly used to form the feature vector.

Color descriptors that inculcate color spatial distribution include the color coherence vector (CCV) [70], the border/interior pixel classification (BIC) [91], and the color correlogram [35].

2.2.2 Texture Features

A texture may be described by the presence of basic primitives, whose spatial arrangement creates some visual patterns expressed in the form of granularity, directionality, and repetitiveness [97]. Textures are not as well defined as color features. However, they

provide valuable information in image classification, as they encode the content of many real-world pictures (e.g., clouds, fruit skin, trees, and cloth) [43]. Texture is hence an important feature for image retrieval purposes.

Different approaches to extracting and representing textures exist. They may be classified as space-based models, frequency-based models, and texture signatures [17]. Some of the techniques used to represent texture are described below.

The co-occurrence matrix [28], which is one of the most traditional methods for encoding texture information, describes spatial relationships among the gray levels in an image. A cell defined by the position (i, j) in this matrix registers the probability at which two pixels 'i' and 'j' containing gray levels occur in positions relative to each other. The literature has proposed a set of co-occurrence probabilities (such as entropy, energy, and contrast) to characterize textured areas.

The edge histogram descriptor (EHD), another space-based texture representation method, is efficient for representing natural images [78]. It captures the spatial distribution of edges, to a certain extent in the same manner as the color layout descriptor. To compute the EHD, a given image is first divided into 4 X 4 sub-images, after which the local edge histograms for each sub-image are calculated. Edges are broadly grouped into five categories: vertical, horizontal, 45°, 135°, and neutral. As a result, each local histogram in one region has five bins corresponding to the above five categories. An image that is divided into 16 sub-images generates 80 bins in total. These bins are quantized in a non-uniform manner using 3 bits/bin, which creates a 240-bit descriptor. The limitation is that the EHD can be very sensitive to object or scene distortions [78].

Frequency-based texture descriptors include inter alia the Gabor wavelet coefficients [48]. The texture signature characterizes texture information on contrast, coarseness, and directionality. An illustration of texture signatures can be found in [93].

2.2.3 Shape Features

Shape features are important to CBIR systems, although they are not as widely used as color and texture [43]. According to the authors in [43], "*for color images used in most papers, however, it is difficult to apply shape features compared to color and texture due to the inaccuracy of segmentation.*" Shapes, which are often determined by applying segmentation or edge detection to an image, commonly include aspect ratio, Fourier descriptors, circularity, moment invariants, and connected boundary fragments [60]. Shape features are found to be useful in some specific domains that contain man-made objects [43].

2.2.4 Spatial Location

In addition to color, texture, and shape features, spatial location is also beneficial in region classification. For example, 'ocean' and 'sky' could have the same color and texture features, but their relative spatial locations are different. The sky usually seems to be at the top of an image, while the ocean is at the bottom. In [47], a region centroid and its minimum bounding rectangle are used to provide spatial location information. In [62], a spatial center of an area is used to depict feature's spatial location.

2.2.5 Discussion

Table 1 below summarizes various color spaces that are commonly used. The LAB color space is a linear color space, whereas the RGB, CMYK, HSV, and YCrCb models are non-linear; as such, the LAB color space a more suitable choice for color manipulations. The RGB model is commonly used in video displays, as it is easy to use and understand; CMYK is widely used in color printers. These two models have some common drawbacks, i.e. they are device dependent and non-linear. The advantage of HSV is that it is easy to understand and calculate, although it is non-linear and does not provide real insight into color production. The YCrCb color space is used in jpeg images to provide greater compression without a significant effect on perceptual image quality. Finally, the LAB color space offers a wider range than RGB and CMYK. It is also device independent and follows a linear

function, which makes it more suitable for manipulations. Its one weakness is that it requires knowledge of color temperature scale.

Table 1. Color space summary [12]

Color System	Parameters	Pros	Cons
RGB	Red, green, and blue	➢ Excellent for video display. ➢ Additive color properties	➢ Perceived as non-linear. ➢ Device dependent.
CMYK	C = cyan M = magenta Y = yellow K = key (black)	➢ Used in printers.	➢ Perceived as non-linear. ➢ Device dependent.
HSV	H = hue S = saturation V = value or lightness	➢ Easy to understand and calculate.	➢ Perceived as non-linear. ➢ Provides no real insight into color production or manipulation.
YCrCb	Y = luminance Cr = red to green Cb = blue to yellow	➢ Excellent for image compression. ➢ Extensively used in image file formats to save space.	➢ Perceived as non-linear. ➢ The color temperature must be known.
LAB	L = luminance A = red to green B = blue to yellow	➢ Wider range than RGB & CMYK color spaces. ➢ Device independent. ➢ Commonly used in color manipulation software. ➢ Perceived as linear.	➢ Color temperature scale must be known.

Table 2 below shows the advantages and disadvantages of the various color descriptors. Color histograms are widely used because they are easy to calculate and intuitive; their drawbacks are that they lead to high dimensions, do not incorporate a feature's spatial information, and are sensitive to noise. Color moments are compact and robust, but they do not encode spatial information and are incapable of capturing all colors.

Finally, while the CCV and correlogram both store spatial information, they also both suffer from high computation costs.

Table 2. Contrast of different color descriptors [74]

Color Descriptor	Pros	Cons
Histogram	➢ Simple to compute. ➢ Intuitive.	➢ High dimension. ➢ Do not store spatial information. ➢ Sensitive to noise.
Color moments	➢ Compact. ➢ Robust.	➢ Not enough to describe all colors. ➢ No spatial information stored.
CCV	➢ Store spatial information.	➢ High dimension. ➢ High computation cost.
Color correlogram	➢ Store spatial information.	➢ Very high computation cost. ➢ Sensitive to noise, rotation, and scale.

The Mediatoil imagery is different from other domains (such as the Caltech 101 dataset [2], COREL dataset [16], and the letter recognition dataset [99]), in that a single category contains images with a great deal less variation; for instance, only images of book cover pages are included in the 'Books' category. Shape features could be useful when identifying images of similar shapes (such as mugs or boats). In our study each class has semantically related pictures, such as 'Protest' images (i.e. all pictures containing people involved in revolts) and 'Graphics' (all computer-assisted visuals), but they do not have the same shape feature. Texture features are applicable in identifying objects, that follow a particular pattern, within an image. However, as we are considering an image as a whole, texture feature does not accurately define our images. We ultimately used the LAB color space, as it has a wider range of colors than RGB and CMYK, is device independent, and is linear, which makes it suitable for manipulations.

One way to capture features is to use the color parameter of individual or specific pixels from an image, as discussed previously. The other approach is to utilize detectors, which identify useful features based on certain principles. Some of these detectors and descriptors are covered in the next section.

2.3 Detectors and Descriptors

Detectors identify keypoints within an image while descriptors encode those features in a vector form for similarity matching and retrieval [75]. The first question that arises in relation to choosing a feature is “What makes a good feature?” Intuitively, repeatedly recognizable regions in an image form a useful feature. Flat surfaces and edges are considered bad elements, whereas corners and blobs are considered beneficial [75].

Figure 2 below highlights the different types of features and their relevance for image matching. A flat feature is the worst for comparison, as it is perceivable in Figure 2(a) that flat surfaces tend to provide false matches with all of the other similar areas in an image. Choosing edges as a feature, as shown in Figure 2(b), is also not good for feature identification, as sliding the trait up or down will result in false matches with many other areas. Corners and blobs are considered useful features, as they uniquely identify a point; even when the feature patch is moved in x- or y-direction, no other element exactly matches the given point [75].

Figure 2. Extracting different feature types from an image: (a) flat, (b) edge, (c) corner, and (d) blob

As feature point detectors and descriptors are essential to various computer vision applications, they have received considerable attention over the last few decades. Various types of feature detectors and descriptors can be found in the literature, with different definitions for the type of feature point in an image that is potentially interesting [31]. According to the authors in [31], a feature detector should have the following properties to be utilized in computer vision applications:

- Robustness – The feature detection algorithm should be able to detect the same feature locations independent of scaling, rotation, shifting, photometric deformations, compression artefacts, and noise.
- Repeatability – The feature detector must identify the same features of the same scene or object repeatedly under a variety of viewing conditions.
- Accuracy – It should accurately localize the image features, especially for image matching tasks that require precise correspondences to estimate the epipolar geometry.
- Generality – It should detect the elements that can be used in different applications.
- Efficiency – The feature detection algorithm must identify features in new images quickly to support real-time applications.
- Quantity – The feature detector's procedure should be able to detect all or most of the features in an image where the density of detected features should reflect the information content of the picture for providing a compact image representation.
- Invariance – The method used should be invariant to rotation, scale, and affine transformation.

After deciding "What makes a good feature?" the next question that needs to be addressed is "How should a useful feature be formulated mathematically?" A detailed discussion of this issue may be found in [75]. As most of the primary detectors are based on the same theory as described in [75], below we discuss some of the basic considerations involved in choosing a good feature point.

Consider a small block of pixels around a candidate feature location (x, y) in an image I, as in Figure 3 below.

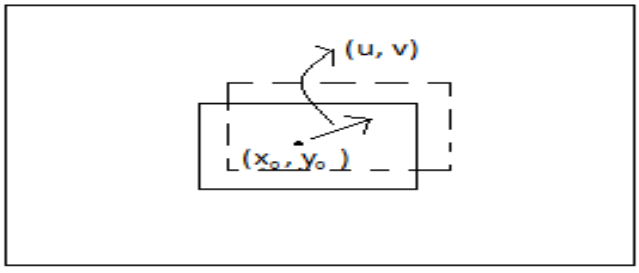


Figure 3. Concept showing a good feature point [75]

$$E(u,v) = \sum_{\substack{(x,y)\in Window \\ centerd\ at\ x_0, y_0}} (I(x,y) - I(x+u, y+v))^2 \quad (1)$$

In Figure 3, (u, v) is a displacement vector. It follows that E (u, v) should be large for any (u, v), for (x_0, y_0) to be a useful feature.

We can also compute how stable the candidate feature x_i,where x_i is a point in (x,y) space, is on small variations in position Δu by examining an image patch I_0against itself, which is also known as the auto-correlation or surface function.

$$E_{AC}(\Delta u) = \sum_i w(x_i)[I_0(x_i + \Delta u) - I_0(x_i)]^2 \quad (2)$$

where $w(x_i)$ is the spatially varying weight function. Expanding $I_0(x_i + \Delta u)$ in a Taylor series yields $I_0(x_i) + \nabla I_0(x_i).\Delta u$. We can thus approximate the auto-correlation function as

$$E_{AC}(\Delta u) = \sum_i w(x_i)[I_0(x_i + \Delta u) - I_0(x_i)]^2 \quad (3)$$

$$\approx \sum_i w(x_i)[I_0(x_i) + \nabla I_0(x_i).\Delta u - I_0(x_i)]^2 \quad (4)$$

$$= \sum_i w(x_i)[\nabla I_0(x_i).\Delta u]^2 \quad (5)$$

$$= \Delta u^T A \Delta u \quad (6)$$

where

$$\nabla I_0(x_i) = (\frac{\partial I_0}{\partial_x}, \frac{\partial I_0}{\partial_y})(x_i) \quad (7)$$

is the image gradient.

The auto-correlation matrix A can be written as

$$A = w * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (8)$$

where the weighted summations are replaced with discrete convolutions with the weighting kernel w.

The detectors are mostly based on the above equations. Below we discuss various feature detection and description methods in detail.

2.3.1 Detectors

The basic idea of a detector is to identify keypoints from an image such that they are distinct and can repeatedly be found within the given picture [31]. Several detectors are available in the literature; here we discuss the most widely used algorithms in the chronological order by year of creation.

2.3.1.1 The Harris Corner Detector

Harris et al. [29] have developed a combined corner and edge detector by obtaining the variation of auto-correlation over all different orientations, which results in a more desirable detector concerning detection and repeatability rate. The 2 * 2 symmetric auto-correlation matrix may be represented as

$$H(x,y) = \sum_{(u,v)} w(u,v) * \begin{bmatrix} I_x^2(x,y) & I_x I_y(x,y) \\ I_x I_y(x,y) & I_y^2(x,y) \end{bmatrix} \quad (9)$$

where I_x and I_y are local image derivatives in x-direction and y-direction, respectively, and w(u, v) denotes a weighting window over the area (u, v). To find interest points, the eigenvalues of the matrix H are computed for each pixel. Both eigenvalues being large indicates the existence of a corner at that location.

The Harris quality measure C is defined as

$$C = \det(H) - k\ trace(H) \quad (10)$$

where

$$\det(H) = \lambda_1 * \lambda_2, and\ trace(H) = \lambda_1 + \lambda_2 \quad (11)$$

Here k is an adjusting parameter and λ_1, λ_2 are the eigenvalues of the H matrix.

The C value for two eigenvalues for the different feature types is shown in Figure 4 below, while a diagram that illustrates the classification of the detected points is provided in Figure 5. If both eigenvalues are small, the resulting C value will be low (i.e. close to zero) and the feature will be considered a flat region. If one of the eigenvalues is low, the resultant C parameter is negative or small and the feature is classified as an edge. For corners and blobs, both eigenvalues will be large and the corresponding C value will be high, which means there is no direction vector (u, v) and moving in that direction decreases the C value. Corners and blobs are thus considered useful features.

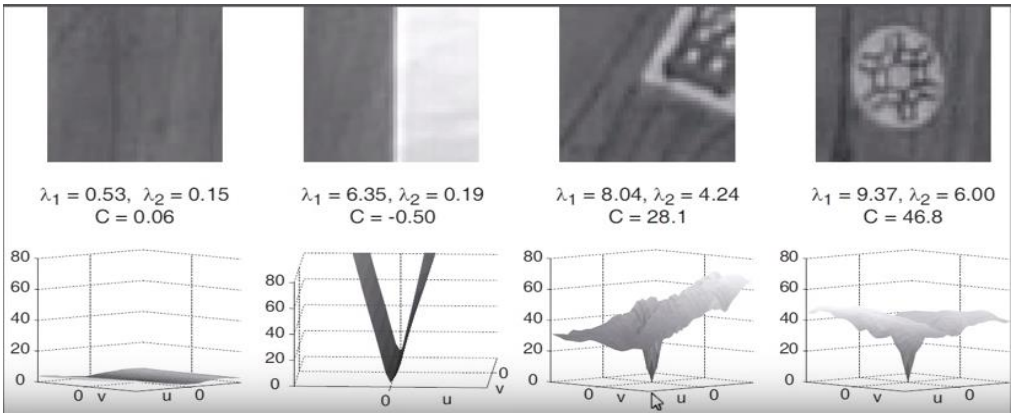


Figure 4. Quality measure C value for flat, edge, corner, and blob type features [75]

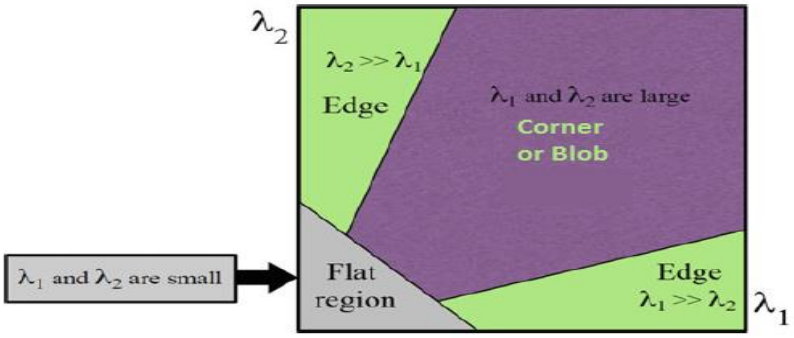


Figure 5. Classification of image points based on the eigenvalues of the autocorrelation matrix H [31]

The Harris method has several advantages: first, it performs better than edge detectors; second, its descriptor is translation and rotation invariant; and third, it significantly reduces the amount of computation compared to tracking every pixel. Its major drawback is that it is not scale independent [38].

2.3.1.2 Scale Invariant Feature Transform

In Section 2.3.1.1, it was noted that the Harris corner detector is not able to detect features scaled at a different level. The scale invariant feature transform (SIFT) approach, on the other hand, is scale and rotation invariant. In [46], Lowe proposes an efficient algorithm based on local 3D extrema in a scale-space pyramid built with difference-of-Gaussian (DoG) filters. The first phase of keypoint detection is to identify locations and scales that can repeatedly be selected under differing views of the same object. Detecting areas that are invariant to image scale change can be accomplished by searching for stable features over all possible levels, using a continuous function of scale known as scale space. The DoG provides a close approximation to the Laplacian-of-Gaussian (LoG) and is used to detect stable features efficiently from scale-space extrema. The DoG function – D(x, y, σ) – can be calculated without convolution by subtracting adjacent scale levels of a Gaussian pyramid separated by a factor k:

$$D(x, y, \sigma) = \big(G(x, y, k\sigma) - G(x, y, \sigma)\big) * I\ (x, y) \tag{12}$$

$$= L(G(x, y, k\sigma) - L(x, y, \sigma) \tag{13}$$

A practical approach for constructing D (x, y, σ) is depicted in Figure 6 below. The images are produced separated by a constant factor k in scale space by incrementally convolving the initial image with Gaussians, as shown on the left side of the figure. Each octave of scale space is then divided, into an integer number, s, of intervals, so that $k = 2^{1/s}$. This produces s + 3 images in the pile of blurry images for each octave, so that resultant extrema detection covers a complete octave. Neighbouring image scales are subtracted to produce the DoG images shown on the right of the figure. Once a full octave has been processed, the Gaussian image that has twice the original value of σ (it will be two images from the top of the stack) is resampled by taking every second pixel in each row and column. The accuracy of sampling relative to σ is no different than that for the start of the previous octave, although computation is significantly reduced.

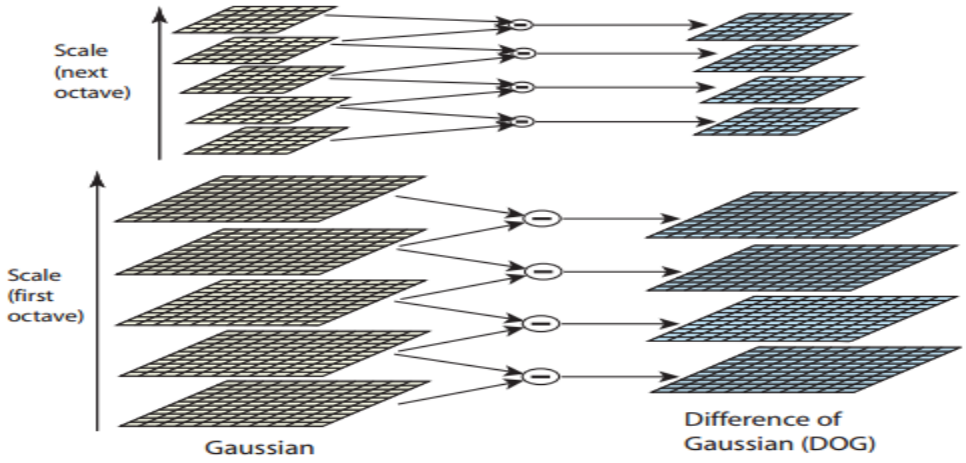


Figure 6. Searching for 3D scale-space extrema in the DoG function [75]

Figure 7 below illustrates that the maxima and minima of the DoG images are detected by matching a pixel (marked with X) to its 26 neighbours in 3 X 3 regions of the current and adjacent scales (marked with circles).

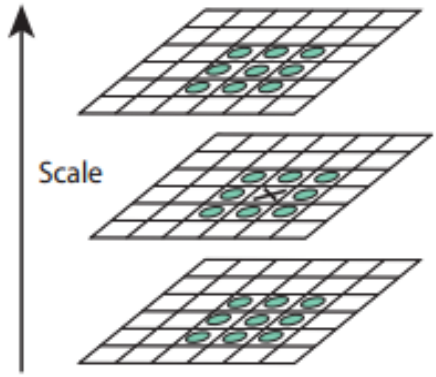


Figure 7. Computation of the maxima and minima of DoG images [75]

The SIFT approach is advantageous as it is independent of scale, translation, and rotation invariance. The common drawback of both its LoG and DoG representations is that the local maxima can also be detected in neighbouring contours of straight edges, where the signal change is only in one direction. It makes them less stable and more sensitive to noise and small changes [64].

2.3.1.3 Maximally Stable Extremal Regions

The maximally stable extremal regions (MSER) approach was proposed by Matas et al. [51] in 2004. The areas are defined exclusively by an extremal property of the intensity function on a region's outer boundary and internal area. The concept of this algorithm can easily be explained using thresholding. In general, to obtain the best feature we want the darkest blob on the lightest background (or vice versa). Assume all possible thresholds of a gray-level image I. All of the pixels below a threshold are marked as 'black', while all of those above or equal to the threshold are labeled as 'white'. For instance, if we consider the threshold images I_t as a continuous movie, we first note a white image and then note some black spots corresponding to local intensity minima. As we increase the threshold, the number of pixels in the region increases. The area of each component (i.e. region) is monitored as the threshold is changed. The regions that have minimal area rates of change with respect to the threshold are defined as maximally stable and returned as detected regions [51].

The MSER approach can be advantageous as it is scale independent, translation and rotation invariant, and ideal for finding vast areas of interest. As stated in [38], another benefit is that MSER's performance is better than that of the Harris method vis-à-vis identifying keypoints. The drawback of MSER is that it prefers round regions [38], which is sometimes not appropriate when discovered sectors are elliptical in shape. Another limitation is the lack of affine invariance in the presence of blur [38], which means MSER is sensitive to changes in rotation and scaling when an image is blurred.

2.3.1.4 FAST Corner Detector

The features from accelerated segment test (FAST) method is a corner detector that was originally developed by Edward et al. in 2006 [76]. Its most significant advantage is that it is computationally efficient, while its primary limitation is that it cannot handle scale variance. As shown in the Figure 8 below, the detector uses a circle of 16 pixels to classify whether a candidate point p is a corner or not. Each pixel in the ring is assigned a label from 1 to 16 in the clockwise direction. If an entire set of N, usually 12, adjoining pixels

in the circle is brighter than the candidate pixel p plus a threshold value t (denoted by I_p + t) or darker than the candidate pixel p minus threshold value t (denoted by I_p - t), p is classified as a corner.

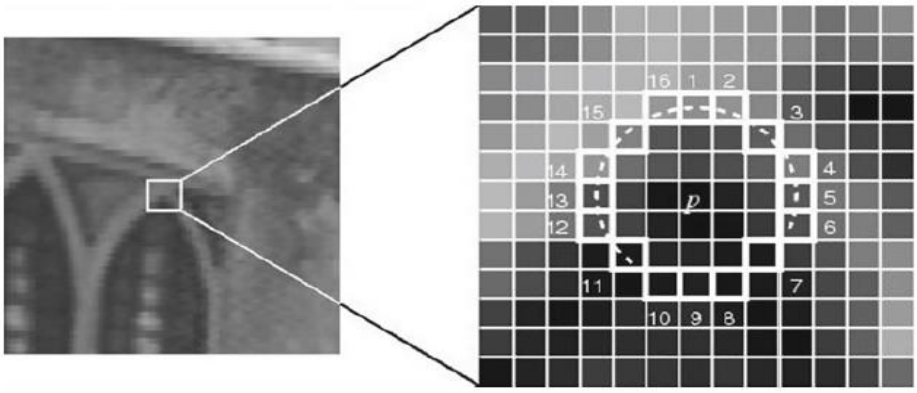


Figure 8. Feature detection in an image patch using FAST detector [76]

A high-speed test can be used to exclude a vast number of non-corner points, by examining only four pixels in question [76]. Pixels 1 and 9 are considered; if both I_1 and I_9 are within [I_p - t, I_p + t], p is not considered as a candidate for the corner. Otherwise, pixels 5 and 13 are further inspected to check whether three of them are brighter than I_p + t or darker than I_p - t. If yes, the rest of the pixels are examined for a conclusion. According to [77], an average of 3.8 pixels must be checked to decide whether a pixel is a candidate for the corner. This significant reduction from 8.5 to 3.8 pixels for each corner candidate could immensely decrease the computation time.

Although a high-speed test leads to low a computation time and high performance, it suffers from several limitations and weakness, as mentioned in [76]. According to the authors in [76], a machine learning approach is introduced to improve the detection algorithm. It operates in two stages. First, corners are detected using the technique discussed above. For candidate p, each location on the circle $x \in \{1, 2, 3, ..., 16\}$ can be denoted by $p \rightarrow x$. The state of each pixel, $S_{p \rightarrow x}$ must be in one of the following three states: darker, similar, or brighter. The set of all pixels of all training images is divided into three different subsets: P_d (darker), P_s (similar), and P_b (brighter).

The second step is to apply a decision tree (DT) algorithm to the 16 locations to achieve the maximum information gain [76]. A recursive process is implemented on each subset to select an attribute x that could maximize the information gain. For example, an x is first selected to partition P into three subsets (namely P_d, P_s, P_b) with the most information gain; another y is then selected for each of these subsets to yield the most information gain (note that y could be the same as x). This recursive process stops when entropy becomes zero to ensure that all pixels in that subset are either corners or non-corners.

The advantage of using the FAST corner detector method is its low computation time, which stems from its use of a high-speed test to discard non-corners. Its disadvantage is that it cannot handle scale changes [76].

2.3.1.5 Speeded-Up Robust Features

The speeded-up robust features (SURF) detector-descriptor scheme, developed by Bay et al. [2], is designed as an efficient alternative to SIFT. In comparison, it is much faster and more robust in finding repeatable keypoints at different scales. For the detection stage of interest points, it does not rely on ideal Gaussian derivatives; the computation is instead based on simple 2D box filters, where it uses a scale invariant blob detector based on the determinant of a Hessian matrix for both scale selection and locations. Its basic idea is to approximate the second order Gaussian derivatives in an efficient way with the help of integral images using a set of box filters. The 9 X 9 box filters depicted in Figure 9 [31] below are approximations of a Gaussian derivative with $\sigma = 1.2$ and represent the lowest scale for computing the blob response maps. These estimations are denoted by D_{xx}, $D_{yy,}$ and D_{xy}. The approximated Hessian determinant can thus be expressed as

$$\det(H_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2 \quad (14)$$

where w is a relative weight for the filter response, which is used to balance the expression for the Hessian's determinant. The approximated Hessian determinant represents the blob response in the image in Figure 9. These responses are stored in a blob response map, and local maxima are detected and refined using quadratic interpolation (as with DoG). Finally,

non-maximum suppression is calculated in a 3 X 3 X 3 neighbourhood to obtain consistent interest points and the scale of values.

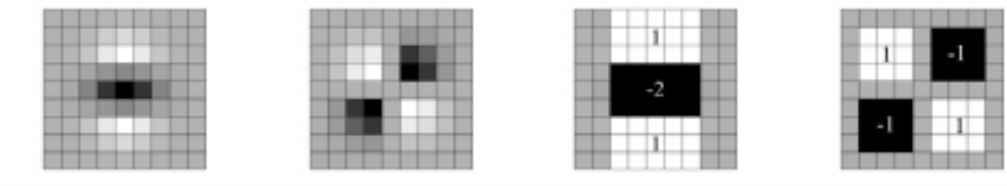


Figure 9. Left to right: the discretized and cropped Gaussian second order partial derivatives in the y-direction (D_{yy}) and xy-direction (D_{xy}), and their approximations using box filters in the same directions, respectively. The gray regions are equal to zero [31].

The major advantage of using the SURF detector is that its descriptor is rotation, scale, and translation invariant [2]. Moreover, it has a small descriptor size in comparison to SIFT, which makes it faster at feature point matching. The downside of SURF and SIFT is that they are not binary, which means they require more time than binary descriptors and are not suitable for mobile platforms.

2.3.1.6 Binary Robust Invariant Scalable Keypoint

Binary robust invariant scalable keypoint (BRISK) is a corner detector that was originally developed by Stefan et al. in 2011 [40]. In contrast to well-established algorithms with proven high performance (such as SIFT and SURF), the BRISK algorithm offers a dramatically faster alternative at comparable matching performance [40]. The BRISK algorithm is an improvement over the FAST detector, where scale invariance is achieved by using the scale space and FAST-score, 's', as measures for saliency. The BRISK detector detects the true extent of each keypoint in the continuous scale space.

Figure 10 below illustrates the scale space representation for detecting scale invariant keypoints. It consists of n octaves c_i and n intra-octaves d_i with 'i' ranging from 0 to n - 1. In an actual implementation of BRISK [40], n = 4. The successive octaves are obtained by half-sampling the image from the previous octave. The intra-octaves are located between octaves c_i and c_{i+1}.

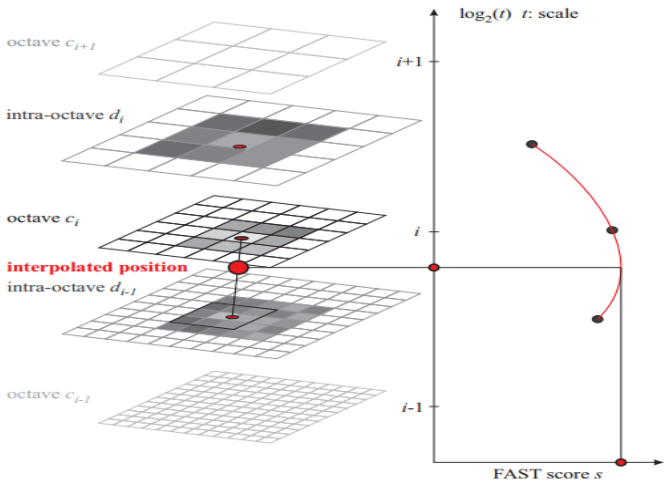


Figure 10. Scale space interest point detection in BRISK [40]

Foremost, the FAST 9-16 detector is applied to each layer of an octave and intra-octave pyramid. Here, 9 out of 16 contiguous pixels are required to be sufficiently darker or brighter than the candidate pixel. Non-maximal suppression is then performed on each layer such that score 's' is maximal within a 3 X 3 neighbourhood and s is also greater than the scales below and above. Using points obtained from non-maximal suppression, a 2D quadratic function is fit to the 3 X 3 patch surrounding the pixel and a sub-pixel maximum is determined. The same is done for the layers above and below. These maxima are then interpolated using a 1D quadratic function across scale space, and the local maximum is chosen as the scale for the feature.

The advantage of using a binary descriptor such as BRISK is that feature matching is fast as calculations are performed by bitwise an XOR operation followed by a bit count on the result, which makes it suitable for mobile platforms [31]. The drawback is that, like other binary descriptors, it is less accurate than vector-based descriptors [31].

2.3.1.7 Discussion

Table 3 below shows that both the Harris and FAST detectors identify corner type features, which are considered good feature points. However, the limitation of these detectors is that they are not scale independent, which makes them less appropriate for finding point correspondences. This means if the same image is matched at different resolutions, the

detectors may fail to match the picture. In contrast, SIFT and SURF, which detect blob-like features, are the most widely used algorithms. The advantage of using these feature detectors is that in addition to being translation and rotation invariant, they are insensitive to scale changes [2, 46].

Table 3. Different detectors along with their feature type and scale independence

Detector	Feature Type	Pros	Cons
Harris corner detector [29]	Corner	➢ Performs better than edge detectors. ➢ Translation and rotation invariant. ➢ Enormously reduces the amount of computation compared to tracking every pixel.	➢ Not scale independent. ➢ Performs below MSER [38].
FAST [76]	Corner	➢ Low computation time due to a high-speed test to discard non-corners.	➢ Not scale independent.
SIFT [46]	Blob	➢ Scale independent. ➢ Translation and rotation invariant.	➢ Less stable and more sensitive to noise or small changes [64]. ➢ Computationally less efficient than SURF due to Gaussian calculation and its 128-bit descriptor.
SURF [2]	Blob	➢ Scale independent. ➢ Translation and rotation invariant. ➢ Computationally more efficient than SIFT due to HAAR wavelets and its 64-bit descriptor.	➢ Not suitable for mobile platforms, due to its non-binary nature.
BRISK [40]	Corner	➢ Scale independent. ➢ Binary features. ➢ Faster computation than other options. ➢ Well suited for mobile devices.	➢ Less efficient in keyword detection than SIFT or SURF. ➢ Less accurate than the vector-based descriptor [31].
MSER [51]	The region with uniform intensity	➢ Scale independent. ➢ Translation and rotation invariant. ➢ Ideal for finding significant areas of interest.	➢ Preference for round regions [38]. ➢ Lack of affine invariance in the presence of blur [38].

The BRISK detector identifies corner features and is also scale invariant; it is faster than SIFT or SURF but is less robust. This detector can be an excellent alternative for low-power mobile devices. The other scale-independent detector is MSER. Although being a rotation, scale, and affine invariant detector makes it a suitable choice for feature detection, MSER is sensitive to blur [38]. Moreover, MSER prefers regular regions [38]; since interesting features in natural images usually have irregular shapes, it is thus inferior to both SIFT and SURF. The SIFT method is computationally expensive, as it involves computing Gaussians for feature detection and takes more time for query matching due to its 128-bit descriptor. The SURF detector's performance is close to that of SIFT; as it only uses a 64-bit descriptor, however, it is a more efficient choice for image feature detection and retrieval.

The task of the detector is to identify keypoints that are yet to be converted to the vectors with the help of descriptors, for later comparison. Below we explore some descriptors that are included in the literature.

2.3.2 Descriptors

Once we have detected interest points from an image at a location p, scale s, and orientation θ, their neighbourhood content needs to be encoded in a suitable descriptor for discriminative matching that is insensitive to local image deformations. A vast number of image feature descriptors are available in the literature; those most frequently used are discussed in the following sections.

2.3.2.1 Histogram of Oriented Gradients

The histogram of oriented gradients (HOG) was first described by Navneet et al. [14] in 2005. It is a very basic descriptor that looks for global features and is mostly used for object recognition. The fundamental idea behind it is that local object appearance and shape within an image can be described by the distribution of edge directions. The HOG descriptor can be best described with the help of an example. Consider a 64 X 128 image. Dividing the image first into 16 X 16 blocks of 50% overlap yields seven blocks horizontally and 15 blocks vertically, for a total of 105 (7 X 15) blocks. Each block is then

further divided into 2 X 2 cells, where the size of each cell is 8 X 8. The next step is to calculate the centered horizontal and vertical gradients with no smoothing and compute gradient orientation and magnitude. The gradient orientations for each cell are then quantized in nine bins (0° to 180°). Each cell is represented by the dominant direction within that cell. Finally, all of the histograms from all of the blocks are concatenated. In this example, the result is a feature vector of length 3,780 (105 X 4 X 9 = 3,780). Figure 11 below shows the edge orientations computed by HOG in a given picture.

Figure 11. A representation of HOG descriptor

2.3.2.2 Local Binary Pattern

Local binary pattern (LBP) [34, 68] is a visual descriptor that has been found to be a dominant feature for texture classification. In its simplest form, LBP is calculated as follows. First, the image is divided into cells (e.g., 16 X 16 pixels for each cell). As illustrated in Figure 12 below, each pixel in a cell is compared its eight neighbours (four horizontals and four diagonals), following in a circular pattern. In its standard version, a pixel c with intensity g_c is labeled as

$$S(g_p - g_c) = \begin{Bmatrix} 1, if\ g_p \geq g_c \\ 0, otherwise \end{Bmatrix} \quad (15)$$

where pixel g_p belongs to one of its eight neighbours. This results in an eight-bit binary number, which is converted to a decimal value for convenience. The next step is to compute the histogram, over the cell, by counting the frequency of each number (ranging from 0 to 255). This histogram can be seen as a 256-dimension vector. Finally, the histograms of all cells are concatenated to define a feature descriptor for the entire window. According to

authors in [94], *"The orientation descriptor of a basic region is defined as the distribution of LBP values in tree channels of CIELab color space of all pixels in the basic region. This distribution is defined as a histogram with 256 bins."*

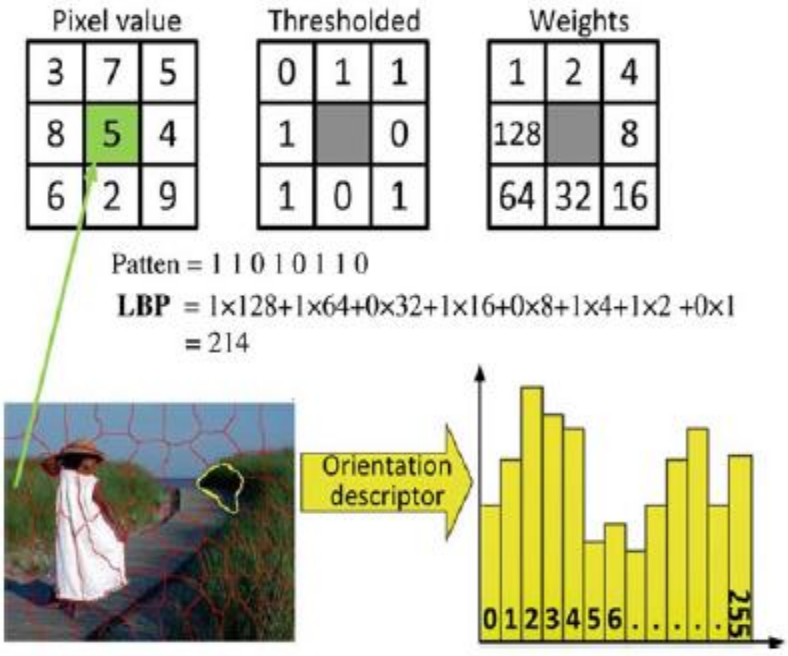


Figure 12. A representation of the LBP descriptor for a pixel in a 3 X 3 neighbourhood [94] © 2014 IEEE

2.3.2.3 Scale Invariant Feature Transform

As noted in Section 2.3.1.2, the SIFT detector identifies interest points using a DoG operator. The next task is to extract feature vectors from these points of interest. The principal concept here is that the histograms of gradient directions must be found from each keypoint in a 16 X 16 window.

As explained in [31], the descriptor stage starts by sampling the image gradient magnitude and orientations in a 16 X 16 area around each keypoint using its scale. The region is then further divided into 4 X 4 sub-regions and a set of orientation histograms is created, in which each sub-region has an eight-orientation bin. The next step is to use the Gaussian weighting function to give more weight to gradients that are closer to the center of the region. Since there are 4 X 4 histograms, each with eight bins, the feature vector

contains 128 (4 X 4 X 8 = 128) elements for each keypoint. Finally, to make it invariant to affine changes, the feature vector is normalized to unit length.

Figure 13 below shows the schematic representation of the SIFT algorithm [31], where the gradient orientations and directions are computed at each pixel and the Gaussian function is the applied (as indicated by the overlaid circle) to give more weights to pixels that are near to the center. A weighted gradient orientation histogram is then computed for each sub-region.

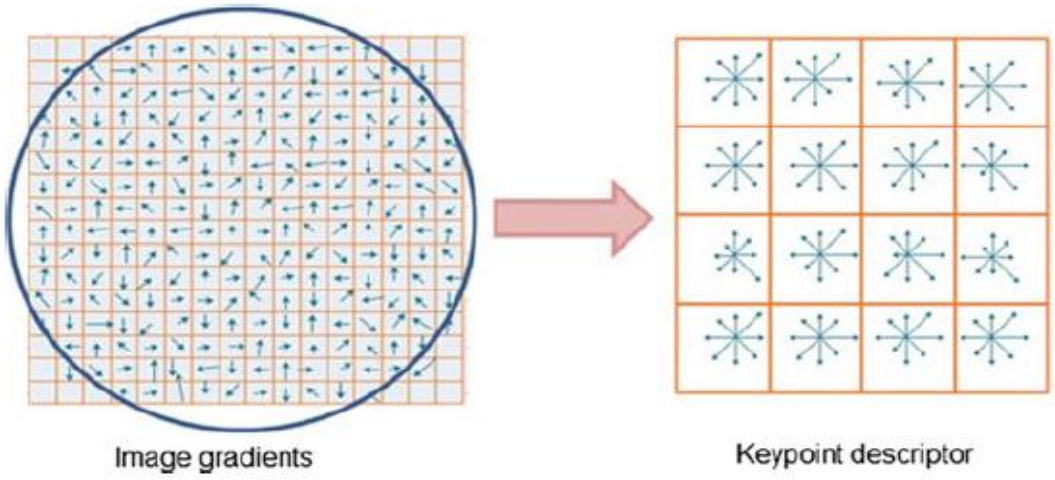


Figure 13. A representation of the SIFT descriptor [31]

The pros of the SIFT descriptor are that it is carefully designed to handle smooth changes in location, orientation, and scale. Its cons are that its construction is complicated [31] and it has high dimensionality (i.e. a feature vector of 128 elements), which makes it unsuitable for real-time matching from a large dataset.

2.3.2.4 Gradient Location and Orientation Histogram

The gradient location orientation histogram (GLOH) was developed by Mikolajczyk and Schmidt [63] in 2005. It is an extension of the SIFT descriptor in which the Cartesian location grid is replaced with a log-polar grid and principal component analysis (PCA) is applied to reduce the descriptor's size.

As shown in Figure 14 below, the GLOH uses a log-polar location grid with eight bins in the angular direction and three bins in the radial direction, where the radius is set to 6, 11, and 15 [31]. It leads to 17 location bins. For each interest point, the GLOH descriptor builds a set of histograms using the gradient orientations in 16 bins, which results in a feature vector of 17 X 16 = 272 elements. Furthermore, by computing the covariance matrix for PCA and choosing the highest 128 eigenvectors, the dimensions of feature vector reduce to 128.

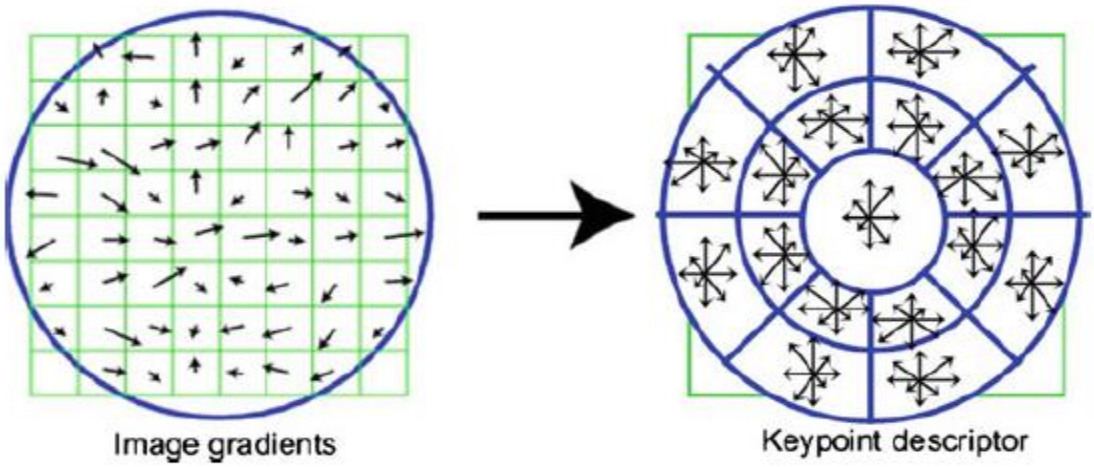


Figure 14. A representation of the GLOH descriptor using log-polar bins [31]

The GLOH method has been shown to be more distinctive than SIFT, although its complexity makes it more expensive to compute.

2.3.2.5 The Speeded-Up Robust Features approach

Section 2.3.1.5 noted that the SURF approach entails detecting blob-like features based on the existing technique of the Hessian blob detector by taking the integer approximation of the determinant of Hessian. The SURF descriptor includes the sum of the HAAR wavelet responses extracted from feature points and their neighbours [3].

The SURF descriptor starts by creating a square region that is centered around the detected interest point and oriented along that point's primary orientation. The size of this window is 20 times the scale at which the interest point is detected [31]. The region of interest is then further divided into smaller 4 X 4 sub-regions; for each sub-region, the

HAAR wavelet responses in the horizontal and vertical directions (denoted as d_x and d_y, respectively) are computed at a 5 X 5 sampled point, as depicted in Figure 15 below. The responses are made more robust against geometric deformations by applying a Gaussian kernel centered at the interest point. The HAAR wavelet responses d_y and d_x are summed up for each sub-region, and a feature vector v is constructed, where

$$v = (\sum d_x, \sum |d_x|, \sum d_y, \sum |d_y|) \quad (16)$$

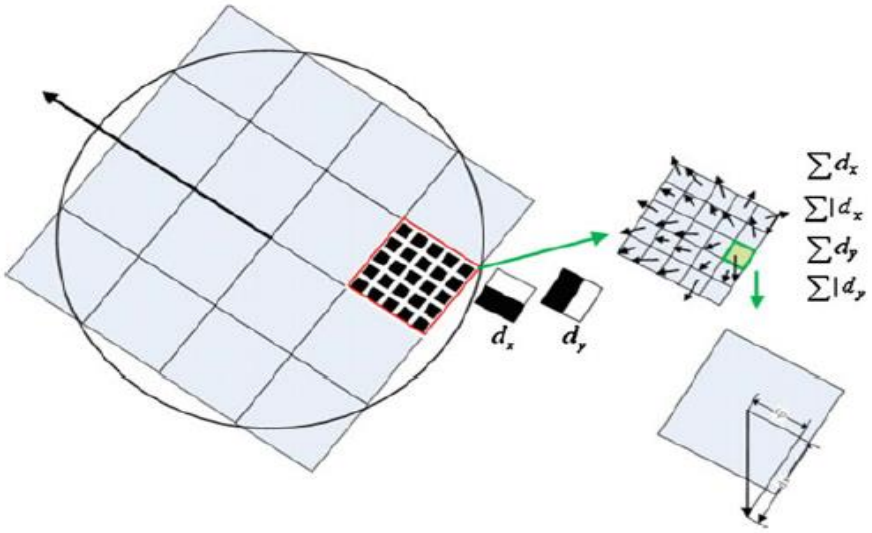


Figure 15. A representation of the SURF descriptor [31]

The result is a feature vector of 64 elements (4 X 4 X 4 = 64). Finally, to reduce illumination effects, the feature vector is normalized to a unit vector.

In brief, SURF is an efficient alternative to SIFT. The SURF descriptor is faster, as it uses only 64 dimensions to describe a local feature in comparison to the 128 elements utilized by SIFT. Moreover, SURF is more robust in producing repeatable keypoints [31].

2.3.2.6 Discussion

Table 4 below provides an overview of the various descriptors discussed in the previous sections. The HOG descriptor is the weakest, as a change in a keypoint's scale or rotation will lead to a mismatch. The LBP method is tolerant to rotation changes, but variations in the size of a keypoint can generate poor results while retrieving images. Both HOG and

LBP fail to find point correspondences within a picture, which means that some keypoints may prevent these two algorithms from finding other relevant keypoints in an image. Both descriptors may be used for classification tasks in which images have to be assigned to a particular category.

Table 4. Different descriptors, including whether they are scale and rotation invariant and their typical use

Descriptor	Invariance		Typical Use		Feature Size
	Scale	Rotation	Finding Point Correspondences	Classification	
HOG	No	No	No	Yes	Depends on image size, as it computes single descriptors globally.
LBP	No	Yes	No	Yes	256
SIFT	Yes	Yes	Yes	Yes	128
GLOH	Yes	Yes	Yes	Yes	128
SURF	Yes	Yes	Yes	Yes	64

In contrast, SIFT, GLOH, and SURF can all handle scale and orientation changes and are well suited for finding point correspondences or classification tasks. As noted, GLOH is an extension to SIFT, although it replaces the Cartesian location grid with a log-polar grid and applies PCA to reduce the descriptor's size. Even though both descriptors have a feature length of 128, GLOH is more expensive to compute. The SURF method, which is best suited for feature point detection, is based on HAAR features that are easy to calculate and takes less time than SIFT or GLOH. Moreover, SURF uses 64-bit descriptor and is more robust in finding keypoints when compared with SIFT. All of these features make SURF a reasonable choice to use as a descriptor in our study.

2.4 Supervised Machine Learning

As explained in Chapter 1, machine learning has been useful for improving a CBIR's retrieval results. A supervised machine learner aims to predict the value of an outcome

measure based on a set of input parameters [43]. The idea is to learn a classifier, such that when classifier is given a feature vector without a class label, the classifier can predict its class.

Several machine learning classifiers, such as naive Bayes (NB), DTs, and support vector machines (SVM), have been used inter alia for object recognition, text classification, regression analysis, and medical imagery classification [96, 114]. For example, in [1], the authors evaluated the performance of five machine learning classifiers: logistic regression, bagging, logistic model trees (LMT), multiclass classifiers (which use logistic classifiers as their defaults), and the attribute selection classifier within the Caltech101 [2] image database. They used the images' texture features and concluded that the logistic regression algorithm was best for classifying the Caltech101 [2] dataset.

In [18], the authors implemented a CBIR for retrieving digital mammograms. They used a two-stage classifier in which the first stage completes a preliminary task to determine whether two images are sufficiently similar for further consideration. Utilizing a two-stage classifier discards some of the relevant images for a query with non-zero probability. They experimented with SVMs and neural networks and considered four models: Fisher-SVM, SVM-SVM, MSVM-SVM, and MSVM-GRN. Their work ultimately showed that MSVM-SVM (i.e. a linear SVM that uses a modified objective function in the first stage and SVM in the second) achieves the best precision-recall ratio.

In [65], the authors introduced a novel approach to using SVM classifiers along with the K-nearest neighbours (K-NN) method for retrieving magnetic resonance imaging (MRI) brain images. Texture features captured utilized a gray-level co-occurrence matrix. In [27], a five-step approach was followed to build a computer-assisted medical diagnosis tool to detect MRI images. In the first step, a Gabor filter was used to extract the texture features. Extracted features were then indexed into a CBIR module. In the next two phases, SVM was trained on the given features, after which the test data was classified with the trained SVM models. Finally, the relevant images were shown with the help of a K-NN query. An accuracy of 94.12% was achieved in the cited paper.

In [86], the authors addressed the problem of class imbalance in thin-layer chromatography (TLC) image classification. Class imbalance is a problem in machine learning that involves the number of instances in one class being far less than the total number of instances in another class. This problem has gained importance in the last decade, as it has been observed that an imbalanced dataset degrades classification performance [86]. The authors in [86] employed a resampling approach to handle the class imbalance that entailed generating synthetic samples for minority class images and under-sampling the majority class.

We next discuss some of the most promising supervised machine learning algorithms that we have used in our research.

2.4.1 Support Vector Machines

The SVM method was originally used for binary classification [13]. Suppose we have a set of training examples $\{x_1, x_2, \ldots, x_n\}$, where $x_i \subseteq R^d$ along with their class labels $\{y_1, y_2, \ldots, y_n\}$ and $y_i \in \{-1, 1\}$. The task of SVM is to find a hyperplane to separate the data with the maximum margin between the hyperplane and the nearest data points of each class, which is illustrated in Figure 16 below. The points closest to the hyperplane are known as 'support vectors'. If the points are linearly separable, linear SVM should be sufficient. However, for non-linearly separable data, the points need to be projected in some higher space where they can be linearly separated; the kernel function does this projection in higher dimensional space.

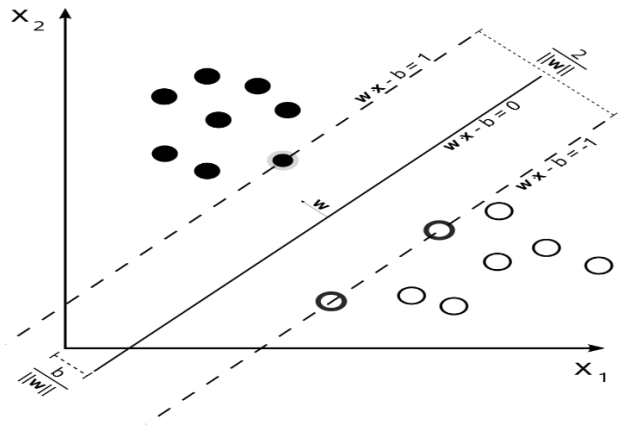


Figure 16. An SVM example of linearly separable data [111]

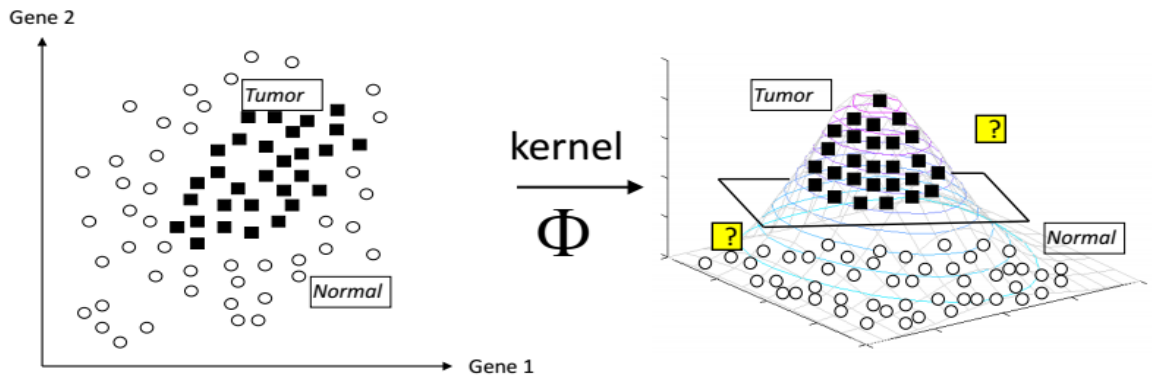


Figure 17. Non-linear data linearly separable in higher dimensional space [90]

The kernel in equation (17) is represented by a dot product and is derived from [90].

$$K(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \,.\, \phi(\vec{x}_j) \tag{17}$$

A linear kernel is defined as $K(\vec{x}_i, \vec{x}_j) = \vec{x}_i \,.\, \vec{x}_j$ and the quadratic kernel is represented as $K(\vec{x}_i, \vec{x}_j) = (p + \vec{x}_i \,.\, \vec{x}_j)^2$ [90]. The quadratic SVM uses the quadratic kernel to divide data linearly in some higher dimension [111].

For multi-class problems, SVM divides a task into multiple binary-class problems. Two approaches are possible: one-vs.-all and one-vs.-one [111]. In the former, one class acts as a positive class and all others work as negative classes. Similarly, n classifiers are trained with each class serving as a positive category. The class that assigns the highest value to the instance is predicted as a class value. In the one-vs.-one method, classification is done using the max-wins voting technique. Here each classifier assigns the query instance to one of the two types. Finally, votes for each class label are counted. The class with the maximum votes assigned to query instance [111].

The main advantage of an SVM classifier is that it can handle high dimensional data [111]. Moreover, the use of the kernel method allows non-linear data to be separated in a linear way by mapping data to some high dimensions [111]. Furthermore, SVM can easily handle numeric and categorical values and works for small training sets [43]. Finally, SVM's generalization performance is high compared to other algorithms [43].

The drawback of SVM is that it requires more time for training models; once the models are prepared, however, the prediction is straightforward. Furthermore, the SVM requires some parameters such as C and r (in the case of a Gaussian kernel) to be carefully chosen for achieving the best classifier [111]. Another disadvantage is that the results cannot be interpreted, as can be done with DTs.

2.4.2 Decision Trees

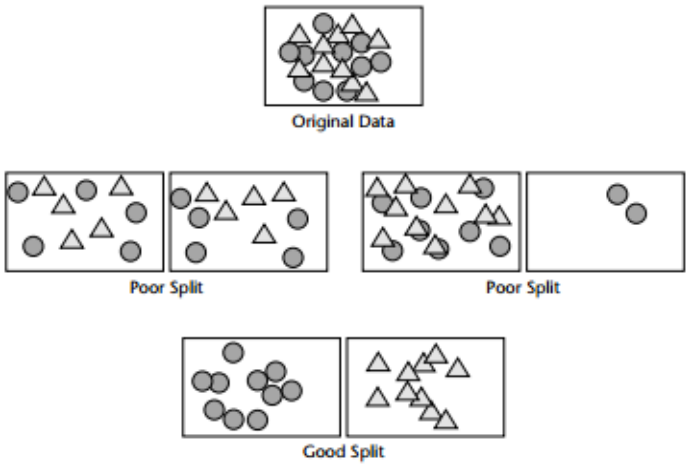


Figure 18. DT: A good split increases purity for all of the children [4]

In addition to the SVM technique, the DT algorithm is also used for learning semantic features. According to the authors in [32], "*DT builds a tree-like structure by recursively partitioning the input attribute space into a set of non-overlapping spaces.*" A DT can also be seen as a set of decision rules from the root to the leaves. It is often used due to its natural interpretation.

A DT is specifically built by dividing the records at each node according to a function of a single input field. The first task is therefore to determine which of the input fields makes the best split. The best division is defined as the one that does the best job of splitting the records into groups where a single class predominates in each group [41].

The authors in [41] define purity as "*the measure used to evaluate a potential split.*" Purity standards can be one of the following: Gini (also known as population diversity), entropy (known as information gain), the chi-square test, incremental response, reduction

in variance, and the F-test. For a complete discussion of purity in relation to DT, readers are recommended to read [41].

Figure 19 below shows the splitting process of a DT. Feature number 2095 of the total 10000 features was chosen as the best candidate for splitting at the root. If the value is less than 0.0085, the left path should be chosen; otherwise, the right path should be selected. If the right path is taken at the root, the image is classified as Protest. At the next level, another feature that increases the purity is chosen. The process continues until the spitting stops increasing the purity or a split is not possible.

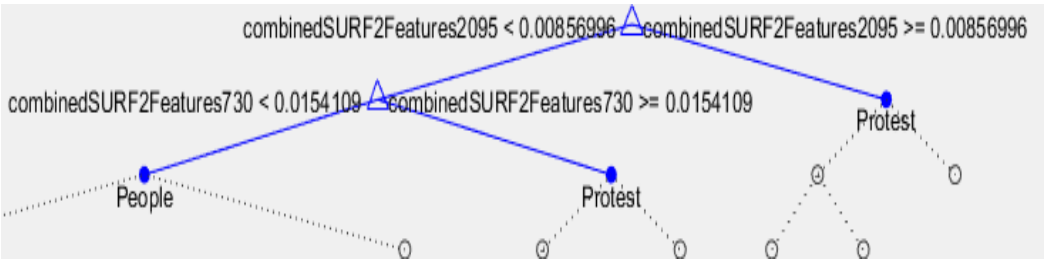


Figure 19. DT: Splitting tree for the People and Protest categories

The main advantage of using DTs is their transparent structure. Unlike other machine learning algorithms, a DT explicitly lists all of the feasible alternatives in a single view. Using a DT is thus suitable when the goal is to generate understandable and explainable rules [41]. Moreover, DTs are also recommended if the data contain numeric or categorical values and the goal is to assign each record to one of a few broad classes. They can also handle missing values by somehow distributing the missing value instances to the child nodes [88].

The main drawback of a DT is that as it broadly categorizes information at each level, it discards some of the information that is available in the training data [41]. As a consequence, another well-tuned classification model can make better use of the data than a DT [41]. Another disadvantage is that DTs are less robust than SVM, as a small amount of noise in the dataset can lead to a completely different tree [112]. One way to eradicate a DT's instability is to use ensemble methods, which are discussed in the next section. According to the authors in [112], "*Decision Trees do not work best if you have a lot of*

uncorrelated variables." Furthermore, diagonal decision boundaries are the hardest to learn in DT and require longer training time given that DTs split data along the x- or y-axis.

2.4.3 Ensemble Methods

Ensemble methods (such as bagging and boosting) use multiple weak learners to form a strong learner that has better predictive performance than individual weak learners [7, 32, 95]. A base learner, which is often referred as a weak learner in ensemble methods, can be any supervised machine learning algorithm (e.g., a DT or a neural network) [115]. It is noteworthy that "*although most theoretical analyses work on weak learners, base learners used in practice are not necessarily weak since using not-so-weak base learners often results in better performance*" [115]. Bagging, which is an abbreviation for bootstrap aggregating, involves several weak learners working in parallel on different randomly drawn subsets of data. In this context, each weak learner votes with equal weight. For instance, a random forest algorithm combines DTs to achieve high classification accuracy [7]. Boosting is a sequential ensemble method in which new models are built on data that previous models misclassified. The most common implementation of boosting is ADABoost [95]. To learn more, readers are recommended to consult [32].

A mutual advantage of bagging and boosting is that they both provide an improvement over the base learner. However, bagging has a few benefits that sometimes make it superior to boosting. First, it reduces the bias and variance by averaging across all models [69]. Second, it achieves better performance than boosting in terms of consistency and area under the curve (AUC) [100]. The AUC is a common evaluation technique for binary classification problems; it is discussed in Section 6.2.2. Finally, bagging also allows parallelized development. Boosting is generally faster than bagging because it grows small trees and works on only the misclassified examples in the next layer. In contrast to bagging, boosting is a linear classifier in which the output of one stage depends on the input of another stage. A disadvantage common to both methods is that they lose the interpretability of the base learner (as a base learner, in the case of the DT).

Table 5. Bagging and boosting algorithms available in Matlab [56]

Algorithm	Regress.	Binary Classification	Binary Classification Multi-Level Pred.	Classify 3+ Classes	Class Imbalance
Bagging	×	×		×	
AdaBoostM1		×			
AdaBoostM2				×	
LogitBoost		×	×		
GentleBoost		×	×		
RobustBoost		×			
LPBoost		×		×	
TotalBoost		×		×	
RUSBoost		×		×	×
LSBoost	×				
Subspace		×		×	

Table 5 above shows the bagging and boosting algorithms available in Matlab, where both ensemble methods use Decision Tree as a base classifier. It is perceived that Bagging is unable to handle the class imbalance problem, whereas RUSBoost is the only boosting algorithm that can handle skewness in classes under current settings.

2.4.4 Weighted K-Nearest Neighbours

In general, K-nearest neighbours (K-NN) is a non-parametric algorithm that is used for classification and regression. Nearest neighbour techniques are quite popular in the domains of fraud detection, customer response prediction, medical treatments, and classifying free-text response [41]. The K-NN method is based on the concept of similarity. It entails searching a database of known records, identifying k-neighbours, and determining the output based on the problem (classification or regression). In classification, the query instance is classified by a major vote of its k-neighbours and the class having the highest number of votes is assigned to the given query. In regression, the output is the average value of its k-neighbours [110].

In the below Figure 20, the value of k is 3. The query instance is represented as a circle (green), and two target categories exist: square (blue) and triangle (red). Based on the voting of k-neighbours, the query is classified as a triangle.

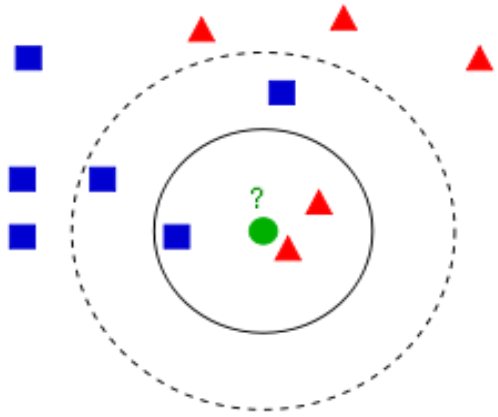


Figure 20. The K-NN concept [110]

Weighted K-NN provides an improvement over regular K-NN. In this method, neighbours' contributions are assigned a weight, so that nearer neighbours contribute more than far neighbours when computing the average or voting of a target class [110].

The K-NN method's advantage over other machine learning algorithms is that merely adding new data to the historical data causes K-NN to learn about new categories [110]. Moreover, the interpretation of the working of K-NN is intuitive.

The disadvantage of K-NN is that it is sensitive to class skewness [110]. One solution for this problem is to weigh the classification, considering the distance from the query point to each of its k-nearest neighbours. The class of each of the k-nearest points is multiplied by a weight proportional to the inverse of the distance from that point to the query point [110]. The other drawback is that it consumes more time than other machine learning techniques, as it has to compare all instances and make a decision at runtime [41].

2.4.5 Discussion

The aforementioned machine learning algorithms are summarized in Table 6 below. These algorithms behave differently in various application domains and have their own pros and cons. As such, it is inappropriate to choose one algorithm over the others without

any experimentation in our domain. This observation reminds us of the "no free lunch" theorem [9], as mentioned in Section 1.2. It is thus interesting to evaluate all of these algorithms on our database and see which algorithm is most suitable when.

Table 6. Advantages and disadvantages of several machine learning algorithms.

Supervised Machine Learning Algorithm	Pros	Cons
SVM	➢ Can handle high dimensional data. ➢ Can handle non-linearly separable data. ➢ Can easily handle numeric and categorical values. ➢ Prediction accuracy is generally high.	➢ Parameter tuning is needed, such as C and r (in case of Gaussian kernel). ➢ Cannot interpret results. ➢ Long training time.
DT	➢ Easy to understand, as each path from the root to a leaf contains a set of rules. ➢ Can handle numeric or categorical values. ➢ Can handle missing values.	➢ Does not completely utilize training data. ➢ Long training time.
K-NN	➢ Intuitive. ➢ Can learn new data easily, by just concatenating in existing database.	➢ Sensitive to the class imbalance problem. ➢ Requires more time for classification or regression.
Bagging	➢ An improvement over the base learner. ➢ Reduces the bias and variance by averaging across all models. ➢ Achieves better performance than boosting in terms of consistency and AUC [100]. ➢ Allows parallelized development.	➢ Bagging a bad classifier can significantly degrade predictive accuracy. ➢ Loses the interpretability of the base learner (a DT, in this case).
Boosting	➢ An improvement over the base learner. ➢ Faster than bagging due to growing small trees. ➢ RUSBoost can handle class imbalance.	➢ A sequential approach. ➢ Loses the interpretability of the base learner (a DT, in this case).

2.5 Summary

In this chapter, we covered the general architecture of CBIR as it forms the foundation for our research. We also discussed various low-level features that are used in CBIR, such as color, texture, shape, and spatial location. We concluded that shape and texture features do not apply to our database. Among various color spaces, the Lab color space is the most appropriate due to its properties of being linear and having a wider range than RGB and CMYK; it is also device independent. The chapter also explored several detectors and descriptors, along with their advantages and disadvantages. In Section 2.4, it was discussed

that corners and blobs are considered good candidates for being keypoints. Among all of the alternatives, the SURF keypoint detector is the best choice given that it is more robust in finding keypoints than any other detector and is scale and rotation invariant. In addition, SURF has a 64-bit descriptor, which makes its computation faster than its counterparts. We also explained that machine learning plays a significant role in building CBIR, by improving the accuracy of results. We discussed various supervised machine learning algorithms (including their pros and cons) and concluded that no single algorithm is best in all domains.

The above conclusion forms the foundation for our research, in which we experiment with the Lab and SURF features on various machine learning algorithms. The aim is to build an efficient CBIR for the Mediatoil. We also investigate the best features to encode the Mediatoil imagery, where our goal is to explore how different stakeholders contribute to the oil sands debate.

The next chapter presents the Mediatoil case study and database, along with the website that we built to explore the database's content containing Canada's oil sands images.

Chapter 3

Mediatoil Case Study

The chapter presents a general introduction to the Mediatoil case study. We also describe the Mediatoil database, which we built as part of this thesis work. Finally, select snapshots from the database are shown to explore the documents available in the Mediatoil dataset.

3.1 Overview of the Mediatoil Project

Mediatoil is a research project that is funded by an Insight Development Grant (IDG) awarded by Canada's Social Sciences and Humanities Research Council (SSHRC) in 2014. The author of this thesis served as the primary computer science research assistant for this project. The project itself is:

> *"interested in how the representational struggle over Alberta's oil sands has been covered out in the publications and campaigns of select oil sands stakeholders. Starting from the oil sand's commercial development in the late 1960's, the Mediatoil project offers a detailed historical analysis of text and images produced in public documents and campaigns by select industry, government, and civil society actors. Of interest are the attributes of these competing media representations, how these representations have developed over time, and how they underwrite the contemporary discursive and visual conventions of the oil sands debate" [59].*

Mediatoil pays particular attention to the oil sands visuals and texts produced by key stakeholders, which are of interest as the public relations and corporate materials produced

around the oil sands provide key resources for the public to understand the debate. It is both fruitful and interesting to look at such texts from a historical perspective in order to understand the context of the debate, its evolution, and its attributes [59].

3.2 The Mediatoil Database

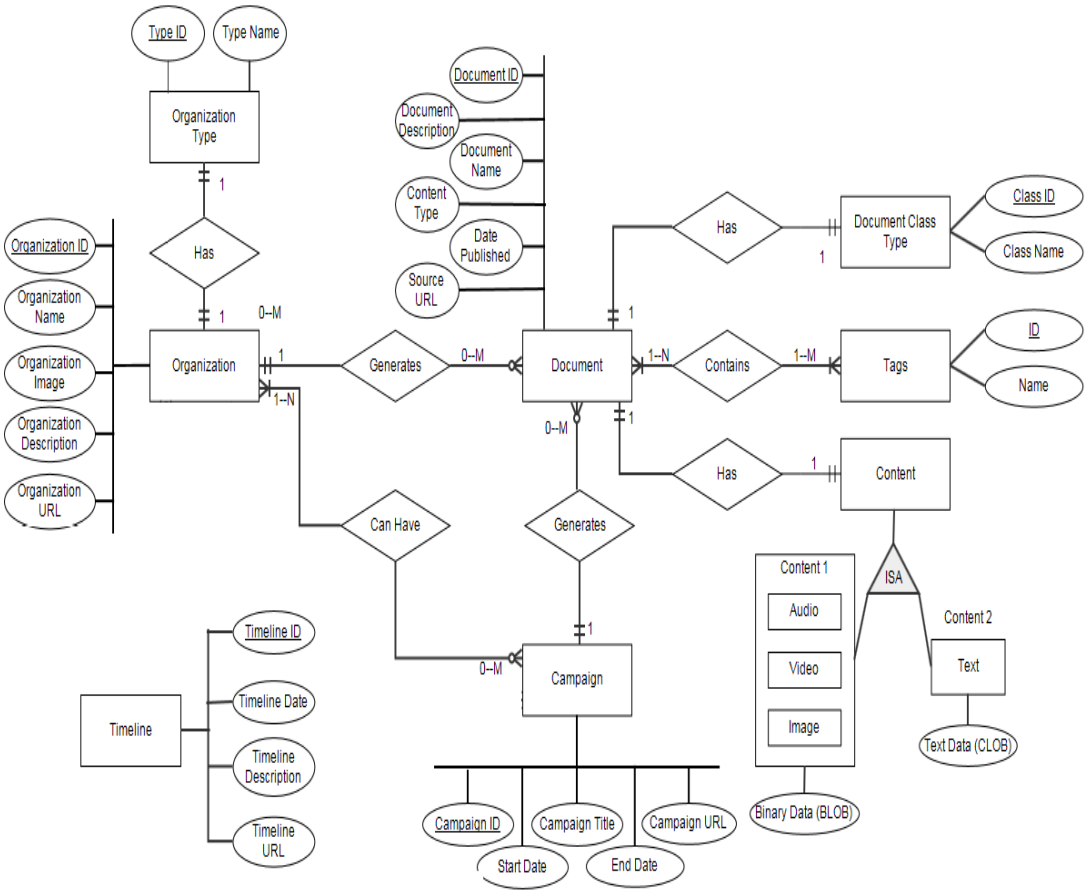


Figure 21. ER diagram of the Mediatoil database

In the previous section, we explained that the Mediatoil project aims at analyzing visuals related to the oil sands as produced by relevant stakeholders. No central repository of tar sands images is available in the literature. To accomplish our objective of analyzing oil sands images, we thus designed and implemented the Mediatoil database,[1] which may be

[1] As the Mediatoil dataset was built from scratch, some ads and campaigns maybe have been missed in the process. Moreover, data collection ended December, 2015. However, we aimed for data collection to be as rigorous as possible in gathering the data.

accessed at www.mediatoil.ca. The entity relationship (ER) diagram is given in Figure 21 above.

There are three primary entities in the Mediatoil database: Organizations, Campaigns, and Documents. An organization is a single stakeholder that has created oil sands imagery. The database contains an ID, name, image, description, and URL for each of the 99 organizations it includes. Another primary entity is a Campaign. Each organization can have one or more campaigns associated with it. Every campaign is assigned an ID, title, start date, end date, and URL in the database; a total of 115 campaigns are included. The third leading entity is a document, by which we mean the images, videos, and reports that we analyzed. Each document has an ID, name, description, content-type, publication date, source URL, and binary content. A total of 2800 records are included, and each document has its binary data stored in a 'BinaryContent' database table (i.e. the documents are stored in their binary form in the database).

The four secondary entities in the database are OrganizationType, DocumentClassType, Tags, and Timeline. The stakeholders were classified into the following organization types: Aboriginal People, Civil Society Pro-Oil Sands, Civil Society Anti-Oil Sands, Industry, Provincial Government, and Federal Government. Each document in the dataset was assigned to one of the following document class-types: Report, Photograph, Still Advertisement, Fact-Sheet, and Video. Furthermore, each document can also have some associated tags. Finally, the timeline is an independent entity that keeps track of important events in the oil sands debates.

Table 7 below shows the 99 distinct stakeholders included in the Mediatoil database along with the total number of documents associated with each organization. It also displays the number of campaigns undertaken by different organizations and the stakeholder group to which a particular organization belongs. The first column shows which of the above-mentioned six organization types each organization has been assigned to. The "total number of documents" column indicates how many documents (including images, videos, and PDFs) related to the respective organization.

Another feature of the database is that it demonstrates the key players in each group. In this context, key players are the organizations that have produced the most documents in any stakeholder group. For instance, in the Aboriginal Peoples group, the primary stakeholder is the Athabasca Chipewyan First Nation (which has produced nine documents); in Civil Society (Anti-Oil Sands), the main organization is Environmental Defence (which is responsible for 162 documents).

Although videos can be viewed using the Mediatoil website, they were not included as part of the analysis. This was because our analysis was concerned with segregating and analyzing static images or report covers into six categories (namely People, Machines, Landscape, Open-Pit, Protest, and Graphics) based on their content. Videos were beyond the scope of the current project, as each frame may show different content; however, they could be considered in future analysis.

Table 7. Each stakeholder's group and number of documents

Stakeholder Group	Stakeholder	Total No. of Campaigns	Total No. of Docs
Aboriginal Peoples	Athabasca Chipewyan First Nation	0	9
Aboriginal Peoples	Coastal First Nations	1	6
Aboriginal Peoples	Mikisew Cree First Nation	0	1
Aboriginal Peoples	Beaver Lake Cree Nation	0	1
Civil Society (Anti-Oil Sands)	Environmental Defence	17	162
Civil Society (Anti-Oil Sands)	350.org	1	87
Civil Society (Anti-Oil Sands)	Indigenous Environmental Network (IEN)	1	79
Civil Society (Anti-Oil Sands)	UK Tar Sands Network	8	73
Civil Society (Anti-Oil Sands)	ForestEthics	4	61
Civil Society (Anti-Oil Sands)	Greenpeace Canada	3	54
Civil Society (Anti-Oil Sands)	The Pembina Institute	6	51
Civil Society (Anti-Oil Sands)	Sierra Club Canada Foundation	6	36
Civil Society (Anti-Oil Sands)	West Coast Environmental Law	0	33
Civil Society (Anti-Oil Sands)	Council of Canadians	1	27
Civil Society (Anti-Oil Sands)	Tar Sands Blockade	1	24
Civil Society (Anti-Oil Sands)	WWF Global	2	24
Civil Society (Anti-Oil Sands)	Great Plains Tar Sands Resistance	1	19
Civil Society (Anti-Oil Sands)	Natural Resources Defense Council	2	15
Civil Society (Anti-Oil Sands)	National Wildlife Federation	1	13
Civil Society (Anti-Oil Sands)	DirtyOilSands.org	0	9
Civil Society (Anti-Oil Sands)	Bold Nebraska	0	8
Civil Society (Anti-Oil Sands)	WWF-Canada	0	8

Civil Society (Anti-Oil Sands)	SumOfUs.org	1	7
Civil Society (Anti-Oil Sands)	Polaris Institute	1	7
Civil Society (Anti-Oil Sands)	People & Planet	2	7
Civil Society (Anti-Oil Sands)	Corporate Ethics International	2	7
Civil Society (Anti-Oil Sands)	Équiterre	1	6
Civil Society (Anti-Oil Sands)	Prince Rupert Environmental Society	2	5
Civil Society (Anti-Oil Sands)	Tar Sands Solutions Network	0	5
Civil Society (Anti-Oil Sands)	Rainforest Action Network (RAN)	1	4
Civil Society (Anti-Oil Sands)	Oil Change International	0	4
Civil Society (Anti-Oil Sands)	Defend Our Coast	0	4
Civil Society (Anti-Oil Sands)	Canadian Parks and Wilderness Society (CPAWS)	1	4
Civil Society (Anti-Oil Sands)	Earthworks	1	3
Civil Society (Anti-Oil Sands)	Friends of the Earth Europe	0	3
Civil Society (Anti-Oil Sands)	Moms Clean Air Force	0	3
Civil Society (Anti-Oil Sands)	Water Matters Society of Alberta	0	3
Civil Society (Anti-Oil Sands)	Stop the Looting	1	2
Civil Society (Anti-Oil Sands)	Natural Resources Council of Maine	1	2
Civil Society (Anti-Oil Sands)	Pipe Up Against Enbridge	0	2
Civil Society (Anti-Oil Sands)	Nobel Women's Initiative	0	2
Civil Society (Anti-Oil Sands)	Keepers of the Athabasca	1	2
Civil Society (Anti-Oil Sands)	Climate Parents	0	2
Civil Society (Anti-Oil Sands)	Desmog Canada	0	1
Civil Society (Anti-Oil Sands)	Ecology Action Centre	0	1
Civil Society (Anti-Oil Sands)	Dogwood Initiative	0	1
Civil Society (Anti-Oil Sands)	Boreal Songbird Initiative	0	1
Civil Society (Anti-Oil Sands)	Canadian Boreal Initiative	0	1
Civil Society (Anti-Oil Sands)	Alberta Research Council	0	1
Civil Society (Anti-Oil Sands)	Alberta Wilderness Association	0	1
Civil Society (Anti-Oil Sands)	Living Oceans	0	1
Civil Society (Anti-Oil Sands)	Friends of the Earth U.S.	0	1
Civil Society (Anti-Oil Sands)	Eurosif	0	1
Civil Society (Anti-Oil Sands)	Pipeline Safety Trust	0	1
Civil Society (Anti-Oil Sands)	Sierra Club Maine	1	1
Civil Society (Anti-Oil Sands)	Saskatchewan Environmental Society	1	1
Civil Society (Anti-Oil Sands)	Western Resource Advocates	0	1
Civil Society (Anti-Oil Sands)	Tides Canada	0	1
Civil Society (Anti-Oil Sands)	Transportation & Environment	0	1
Civil Society (Pro-Oil Sands)	EthicalOil.org	7	30
Civil Society (Pro-Oil Sands)	Canada Action	4	21
Civil Society (Pro-Oil Sands)	Canadian Centre for Energy Information	0	2
Civil Society (Pro-Oil Sands)	Canadian Energy Systems Analysis Research	0	1
Federal Government	Government of Canada	2	38
Federal Government	Natural Resources Canada	1	11

Federal Government	The National Academies of Sciences, Engineering, and Medicine	1	1
Industry	Suncor Energy	4	430
Industry	TransCanada	7	190
Industry	Canadian Association of Petroleum Producers (CAPP)	5	180
Industry	Syncrude Canada	2	131
Industry	Shell Canada	1	114
Industry	Enbridge	5	95
Industry	Canadian Energy Pipeline Association (CEPA)	4	94
Industry	Cenovus Energy	7	81
Industry	Canada's Oil Sands Innovation Alliance (COSIA)	0	72
Industry	Canadian Oil Sands Limited	0	14
Industry	Canadian Oil Sands Trust	0	13
Industry	Encana Corporation	0	7
Industry	Imperial Oil	0	6
Industry	Young Pipeliners Association of Canada (YPAC)	0	6
Industry	BlackRock Ventures Inc.	0	5
Industry	Canadian Natural Resources Limited	0	4
Industry	CS Resources Limited	0	4
Industry	Deer Creek Energy Limited	0	2
Industry	Gulf Canada Limited	0	2
Industry	The Co-operative	0	2
Industry	Total	0	2
Industry	Canadian Occidental Petroleum Ltd.	0	2
Industry	Athabasca Oil Sands Trust	0	2
Industry	Bechtel Canada	0	1
Industry	Canadian Fuels Association	0	1
Industry	Lush	0	1
Industry	Nexen	0	1
Industry	Suncor (Petro-Canada)	0	1
Industry	Enform	0	1
Industry	Connacher Oil & Gas Limited	0	1
Provincial Government	Government of Alberta	3	389
Provincial Government	Government of Alberta - Alberta Energy	1	58
Provincial Government	Government of Alberta - Oil Sands Discovery Centre	0	15

Table 8 below highlights the number of organizations in each stakeholder group. Each stakeholder, grouped into a generic category, can have several documents related to different document class-types (namely Still Advertisements, Reports, Photographs, and Factsheets/Graphics).

Table 8. Number of documents within each stakeholder group bifurcated by document class-type (Still Advertisements, Reports, Photographs, Factsheets/Graphics)

Stakeholder Group	No. of Stakeholders/ Organizations	Document Class-Type	No. of Images and PDFs
Aboriginal Peoples	4	Still Advertisements	9
		Reports	1
		Photographs	0
		Factsheets/Graphics	1
Civil Society Pro-Oil Sands	4	Still Advertisements	7
		Reports	0
		Photographs	25
		Factsheets/Graphics	10
Civil Society Anti-Oil Sands	55	Still Advertisements	87
		Reports	73
		Photographs	250
		Factsheets/Graphics	90
Federal Government	3	Still Advertisements	6
		Reports	12
		Photographs	23
		Factsheets/Graphics	0
Provincial Government	3	Still Advertisements	15
		Reports	36
		Photographs	149
		Factsheets/Graphics	102
Industry	30	Still Advertisements	77
		Reports	137
		Photographs	423
		Factsheets/Graphics	152

Gathering information regarding stakeholder groups and which document class-type an image belongs to was intrinsic to the database collection process. A second layer of information regarding whether the picture is considered a machine, landscape, or so forth was based on the content of the image. This content-type information was achieved from images after training classifiers on their low-level representations. This extra layer of information allows images to be drilled down up to two levels: first, the document class-type level; second, the content-type level. The entire document classification and image retrieval process is discussed in Chapter 5.

Based on recommendations from Prof. Ptarick McCurdy, Mediatoils' prime investigator (PI), six categories were identified for classifying Mediatoil images. The underlying intention was for the categories to be mutually exclusive. However, it was later

identified that some of the classes do overlap with others. A three-step approach for developing Mediatoil-IR was followed to resolve this overlap, as explained in Section 5.2.2. Each category is discussed and defined below.

- Graphics – Images in this class contain computer-assisted photographs and clip art as well as notices and billboards. This category shares some features with other classes. For example, a graphical poster may also contain Landscape, People, or Open-Pit. In the event of overlap, a picture was classified as Landscape, People, or Open-Pit (based on its content) and not as Graphics.

- Machines – These images comprise pictures related to oil sands infrastructure, such as plants, drills, piping, and factories. Steam-assisted gravity drainage (SAGD) imagery may also be included.

- People – Images in this category include pictures of people, including celebrities and children. Pictures of people at a protest are excluded from this class.

- Landscape – Images in this comprise include pictures of landscapes. These images may feature sky, water, fields, and even people – so long as nature predominates.

- Protest – Images in this category show people protesting with or without posters in their hands. They may also contain police arresting individuals at a protest or other protest-related images. People and Protest images also overlapped, as both contain humans. This overlap was resolved by identifying People during a first step and Protest during a second step, as discussed in Section 5.2.2.

- Open-Pit – Images in this class relate to the process of the open-pit mining of bitumen, including extraction (through mining/shovelling/trucks) and tailing ponds. A total of 26 images that shared properties of both Landscape and Open-Pit were categorized after consulting with the prime investigator (PI), Prof. Patrick.

Key pages from the Mediatoil website are illustrated next.

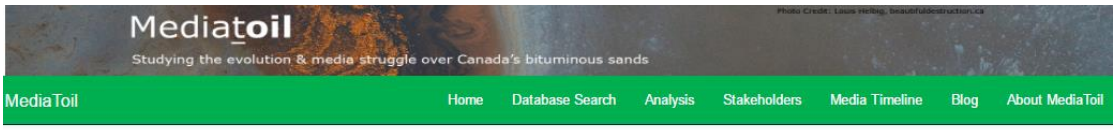
3.3 The Mediatoil Website

The Mediatoil website, which was designed to allow the public to explore the Mediatoil images and database, may be accessed at www.mediatoil.ca. The website was developed in VS2015 with C#, Razor, and Bootstrap frameworks for the front-end, with SQL Server 2014 as a back-end. Creating a website was necessary not only to support our study; we also wanted to make our information open to researchers investigating bituminous sands.

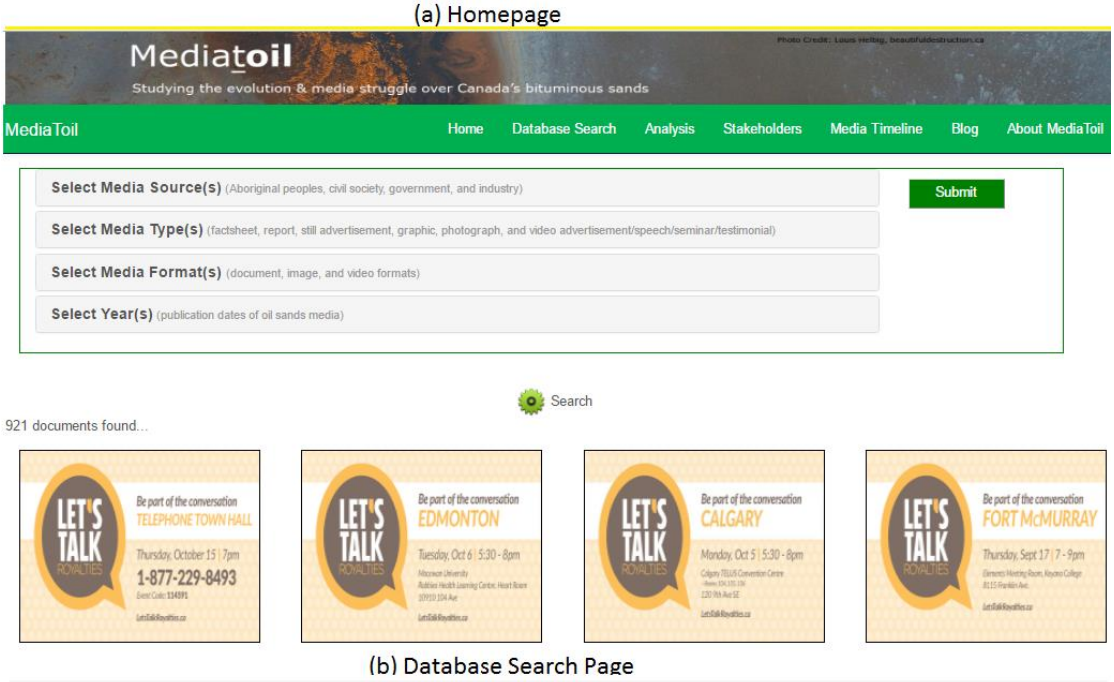
Figure 22(a) below depicts the homepage of the Mediatoil website, which features four main links: Database Search, Stakeholders, Analysis, and Timeline. These links are all discussed in the followings sections.

Figure 22(b) below presents a document search page that allows users to explore the documents available in the database. The page also contains a small 'settings' icon, which enables users to customize their searches by selecting the media source (stakeholder-type), media type (document category), media format (image, video, PDF), and the particular year to search. Once a user has customized the settings, the corresponding search shows all of the relevant documents. The details of each document are easily accessible from this page; users can learn more about, inter alia, a document's source, the actual URL used to retrieve it, publication date, description, and content-type.

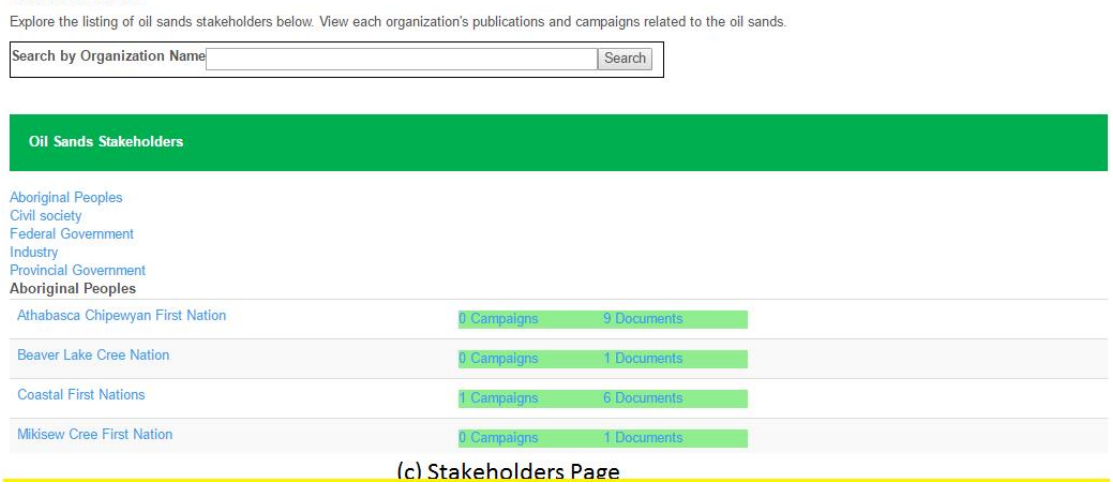
Figure 22(c) below illustrates the stakeholders page, which enables users to access information about different stakeholders in the Mediatoil database. These organizations are categorized into five types: Aboriginal People, Federal Government, Provincial Government, Industry, and Civil Society (pro- and anti-oil sands). This page shows all of the organizations, along with their associated campaigns and total number of documents (which are presented as hyperlinks that lead to details concerning the relevant campaigns and documents).



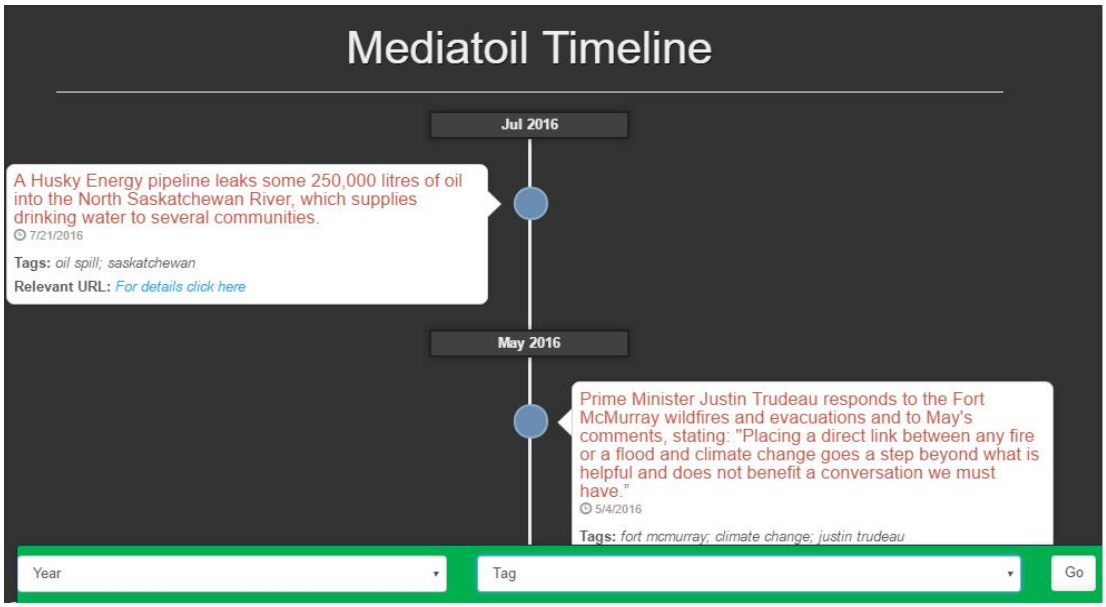
(a) Homepage



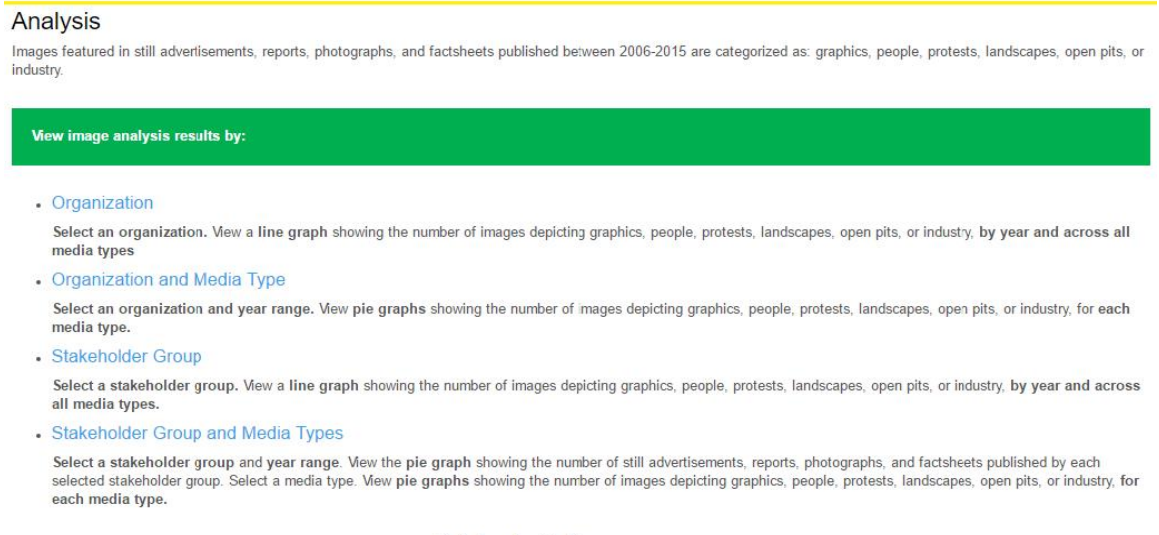
(b) Database Search Page



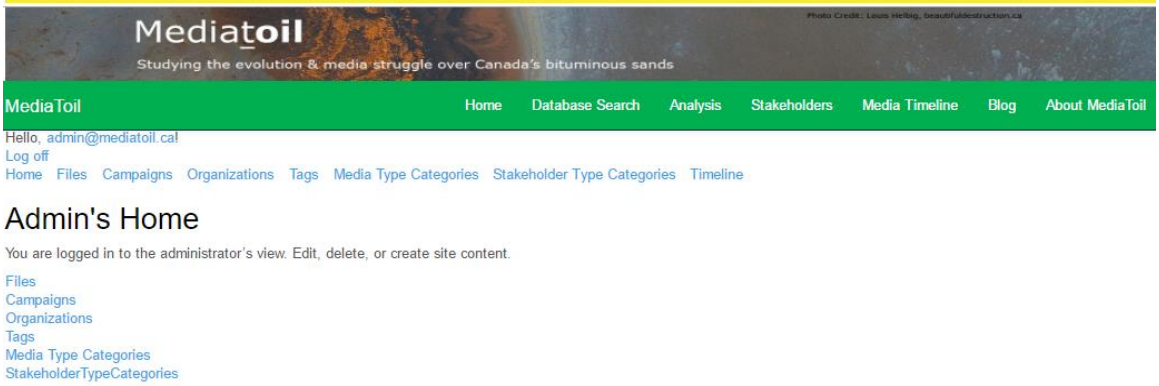
(c) Stakeholders Page



(d) Timeline Page



(e) Analysis Page



(f) Admin Page

Figure 22. The Mediatoil website

Figure 22(d) shows the website's timeline page, which displays a timeline of select key events (e.g., political precedents, protests, campaigns, and news stories) concerning the oil sands and related pipelines. The activities are organized in chronological order from newest to oldest; the earliest recorded news is from November 2004. This page also allows users to search the timeline for events containing a particular tag in a specific year.

Figure 22(e) depicts the analysis page of the Mediatoil database, which contains four links. The first link, 'Organization', allows a user to select an organization and then displays a line graph showing the number of images depicting the six content-types (namely Graphics, People, Protests, Landscape, Open Pits, and Industry) by year and across all media types. The second link, 'Organization and Media Type', allows a user to select an organization and year range and then view pie graphs showing the number of images depicting the six content-types for each media type. The third link, 'Stakeholder Group', allows a user to select a stakeholder group and view a line graph showing the number of images showing the six content-types by year and across all media types. The last link, 'Stakeholder Group and Media Types', allows a user to select a stakeholder group and year range and then view a pie graph showing the number of document content-types (Still Advertisements, Reports, Photographs, and Factsheets) published by each selected stakeholder group. It also allows users to select a media type and view pie graphs that display the number of images depicting the six content-types.

Figure 22(f) illustrates the Mediatoil administrative module, which enables an administrator to edit, delete, and update existing database entries as well as to insert new entries. In particular, an authorized person can access this page to manipulate media, campaigns, stakeholders, tags, stakeholder categories, document types, and timeline.

3.4 Summary

This chapter discussed the Mediatoil case study, the Mediatoil database, and the web portal created to access the dataset. Unlike studies in which data are readily available, the data within our database were collected by extensively searching on- and offline repositories. As we concluded that video analysis is beyond the scope of the current project, we only

analyze the images and cover pages of reports saved as pictures. Finally, some thumbnails of the Mediatoil website were presented to indicate the potential of our work.

In the next chapter, we present the detailed analysis of the imagery used by the different stakeholder organizations.

Chapter 4

Exploring Stakeholder Imagery

In this chapter, we undertake a detailed analysis[2] of how tar sands images from the different stakeholders considered in this study varied. As explained in Section 3.2, the stakeholders in the Mediatoil database are divided into six categories: Aboriginal People, Civil Society Pro-Oil Sands, Civil Society Anti-Oil Sands, Industry, Provincial Government, and Federal Government. Each group can also have four types of documents: Report, Photograph, Still Advertisement, and Fact-Sheet. Furthermore, these images can be classified into six major types based on their content: Open-Pit, Machines, Landscape, People, Graphics, and Protest. The content-type for each image is predicted from trained classification models. Chapter 5 covers in detail, how to train classifiers and predict each images' content-type.

In the following paragraphs, we analyze the six different kinds of organizations and the imagery that they used from 1925 to mid-2016.

The Aboriginal Peoples category contained only 15 documents, which made it easy to analyze by hand. It contained only four organizations: Coastal First Nations contained six (6) documents, Athabasca Chipewyan First Nation generated seven (7) documents, Beaver Lake Cree Nation and Mikisew Cree First Nation produced one document each. Figure 23 below suggests that the Aboriginal Peoples category was against the oil sands. Our further discussion will be focused on other five categories.

[2] The current analysis is based on the information available in the Mediatoil database. The database may not be complete.

Figure 23. Images from the Aboriginal Peoples stakeholder group: (a) Documents produced by Coastal First Nations, (b) documents produced by Athabasca Chipewyan First Nation, (c) documents produced by Beaver Lake Cree Nation, and (d) documents produced by Mikisew Cree First Nation

The pictures from the Industry and Provincial Government categories can be dated back to 1967 and 1925[3] respectively. The year 2004 was a breakthrough year when Civil Society images related to the oil sands came into existence, but we were able to collect only two documents for the period 2004-2005. After manually analyzing these documents, we decided to choose 2006 as a distinguishing year given that this is when Civil Society became more politically active in opposing the oil sands. The two initial documents were entitled "Oil Sands Fever: The environmental implications of Canada's oil sands rush" (Pembina) and "Oil and Gas in British Columbia: 10 steps to responsible development" (West Coast Environment Law). We consequently performed two experiments to see how the images varied in content. The first considered pre-and post-2006 Industry images, while the second looked at pre-and post-2006 Provincial Government photographs. Furthermore, we also presented a time series analysis of different stakeholder groups after 2006.

4.1 Stakeholder Analysis: Industry, Pre- and Post-2006

Figure 24 below compares images in the Industry stakeholder category both pre- and post-2006. Before 2006, stakeholders in this category were mainly producing annual reports using computer-generated graphics to highlight the economic growth figures. In addition to annual reports, a significant increase was observed in Photographs, Factsheets, and Still and Video Advertisements in 2006. When industries started facing opposition from civil society and non-profit organizations after 2006, they started producing documents that show oil as an essential commodity for daily life. Moreover, industries started using their employees to talk about the economic benefits of tar sands and show how responsible they are being to develop the oil sands in a way that minimizes negative impacts on the environment. Figure 25 below shows some images that industry produced after 2006 to try to shape public perceptions of oil sands development.

[3] The earliest available document from 'Provincial Government' was from the year 1925 and was retrieved from the Provincial Government's archive. The picture was captioned "Sidney Blair (right) in 1925, at the site of the experimental bituminous sand separation plant in Edmonton. Source: Provincial Archives of Alberta, A3532" [58].

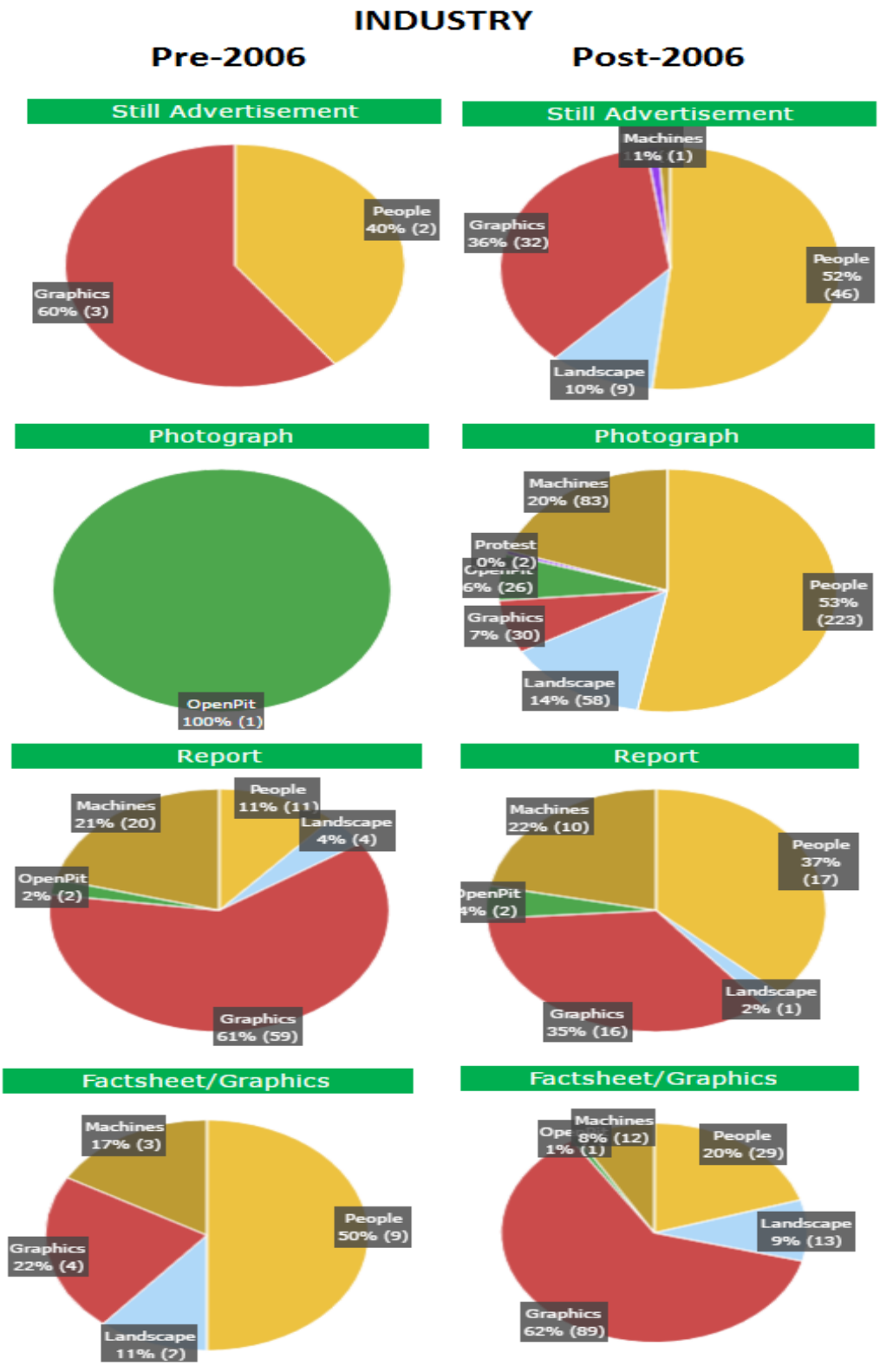


Figure 24. Stakeholder analysis: Industry, pre- and post-2006

Figure 25(a) below contains factsheets/graphics from Enbridge's 2014 'Life takes energy' campaign. Figure 25(b) below presents still advertisements from CAPP's 'OilSandsToday.ca' campaign, which were collected in 2011 and 2012. Figure 25(c) below contains Suncor Energy factsheets/graphics that were gathered in 2009.

Figure 25.Images from the Industry stakeholder group: (a) Oil as an essential commodity, (b) industry employees talking about economic benefits and (c) industry showing its concern for nature

Before 2006, Reports were discovered more than any other kind of images. Most of them contain computer-assisted graphics, and they include annual reports from both Suncor Energy and Shell Canada. In addition to Graphics, images containing Machines are also prevalent. Although Photographs, Still Advertisements, and Factsheets show significant increases in their numbers of documents after 2006, oil sands companies continued to produce Reports to show their profitability.

The number of images belonging to the document-type Photograph shows a significant increase after 2006, in contrast to the earlier period. This increase was due to the evolution of corporate websites. A total of 53% of the documents in this class contain People (mostly employees of the oil sands industry). Nearly 83 images are of Machines and depict work being undertaken to produce oil from the tar sands. As shown in Figure 26 below, approximately 20 pictures contain Open-Pit images from Industry. In some of the pictures, companies are seeking to explain how they intend to return the land to its original state. For instance, Suncor Energy used most of the Open-Pit images in its 'Oil Sands Question

and Response blog (OSQAR)' campaign "*to support constructive dialogue about the oil sands*" [20].

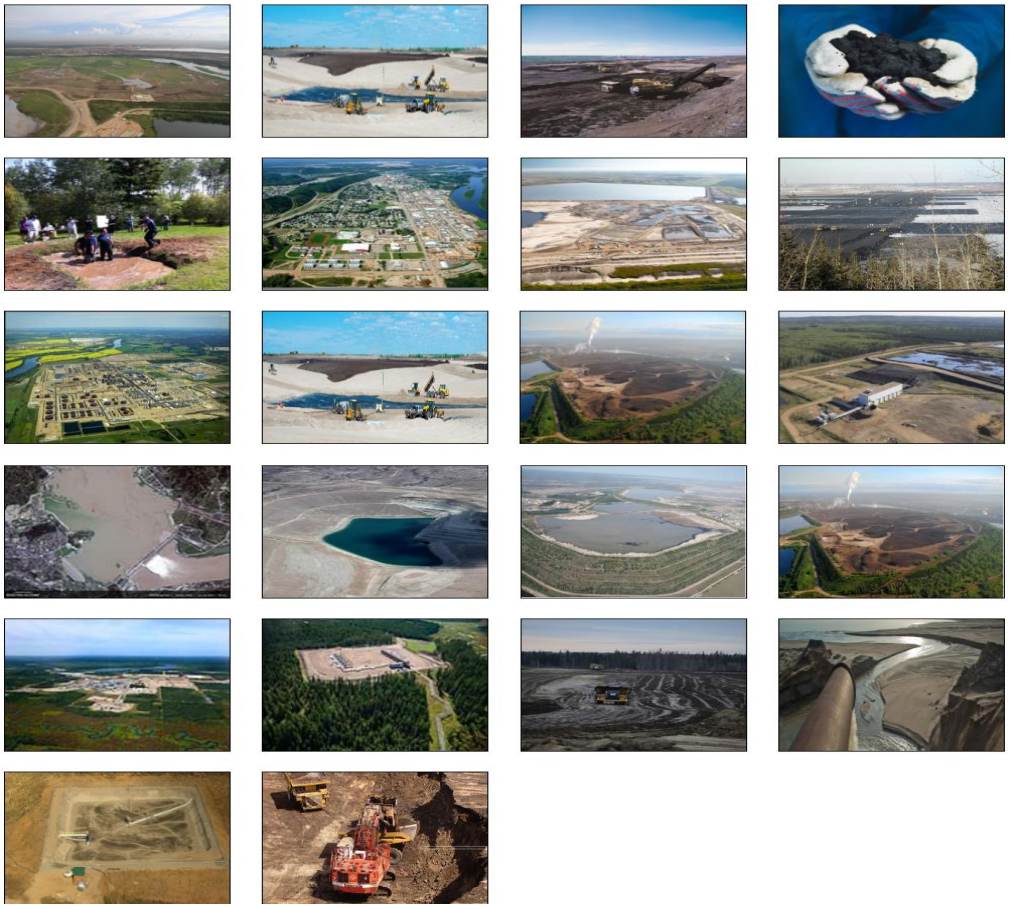


Figure 26. Open-Pit pictures belonging to the Industry document-type

Figure 27. Clip-art pictures belonging to the Industry document-type

Factsheets mainly used People before 2006. A significant portion of Factsheets was later shared by Graphics, as they are easy to produce and can effectively communicate a great deal of information in less space. Approximately 70 clip-art images were used by Industry, but they were impractical for our analysis as shown in Figure 27 above. Most of these images were used by Suncor Energy for its OSQAR campaign, which it launched in 2012 "*to support constructive dialogue about the oil sands*" [20]. This campaign was undertaken on a corporate blog and the images were used to offer additional content and filler for the posts. The images are different from the more traditional "oil sand imagery" that we encountered in other aspects of our project. In addition to Industry, Civil Society Anti-Oil Sands groups also used around 10 clip-art images. Given that the clip-art category was fundamentally different from the environmental images under study, it was excluded from analysis. However, it is notable that a few clip-art images show the oil extraction process and the need for oil mining. A total of 20% of the Factsheets also contained People to show that industry is taking action on water management, land management, and energy being, given that they are essential components of daily life.

Very few Still Advertisements were available from before 2006, but thereafter they show a significant increase. People were most dominant, holding a 62% of this category. Based on the number of documents available, CAPP and Cenovus were the major players; this is shown in Table 9 below. Figure 28 below shows that Graphics were used in Still Advertisements, which industries employed to generate more public support for oil sands. Figure 29 below suggests that Industry also used People in their Still Advertisements to show that oil is essential for driving everything in life.

Table 9. The segregation of images containing People in the Still Advertisements category of Industry

Stakeholder Name	CAPP	Cenovus	TransCanada	Suncor	Shell	Enbridge	CEPA
No. of Documents	15	8	6	6	5	4	3

Figure 28. Graphics pictures belonging to the Still Advertisement document-type in the Industry stakeholder-type

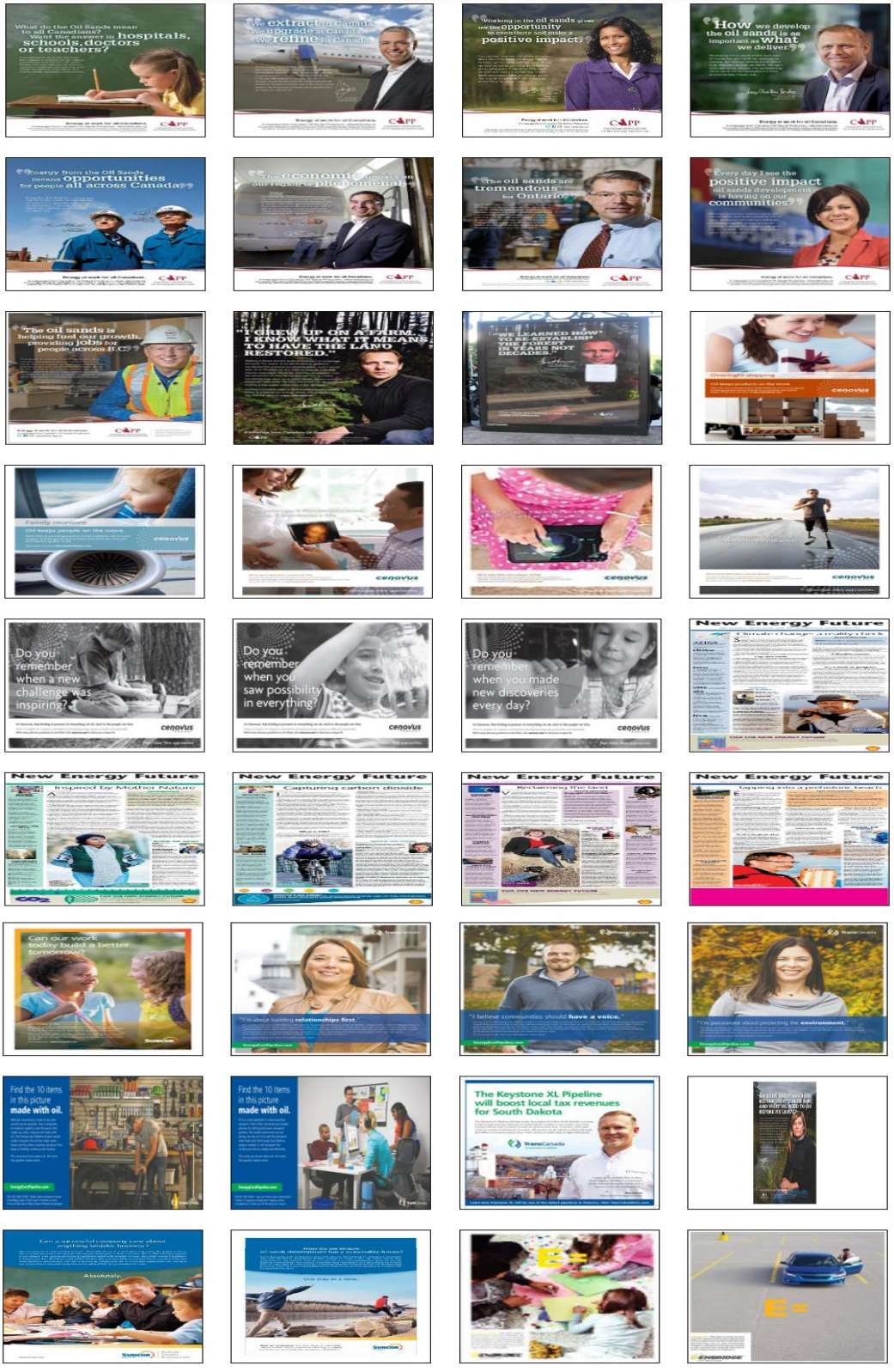


Figure 29. People pictures belonging to the Still Advertisement document-type in the Industry stakeholder-type

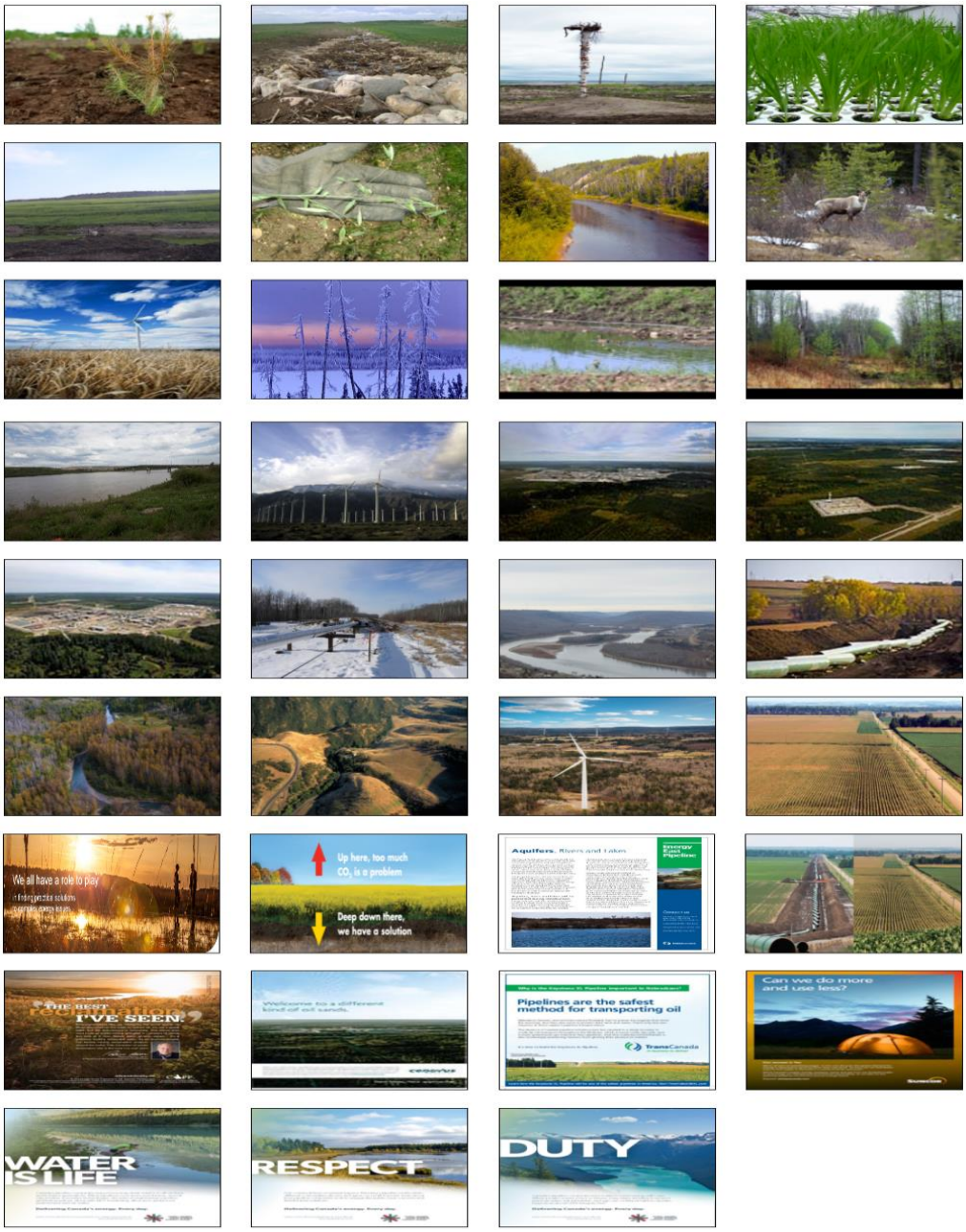


Figure 30. Landscape pictures belonging to the Still Advertisement document-type in the Industry stakeholder-type

As shown in Figure 30 above, industries mostly used Landscape pictures to show how concerned they are about protecting natural resources and returning the land and natural habitat to their original states. Most of these pictures were captured from Suncor Energy's 'Oil Sands Question and Response' campaign, which ran from 2008 to 2015. The Canadian Energy Pipeline Association (CEPA) also produced a few Still Advertisements as part of its 2015 'Delivering Canada's Energy Every Day' campaign.

4.2 Stakeholder Analysis: Provincial Government, Pre- and Post-2006

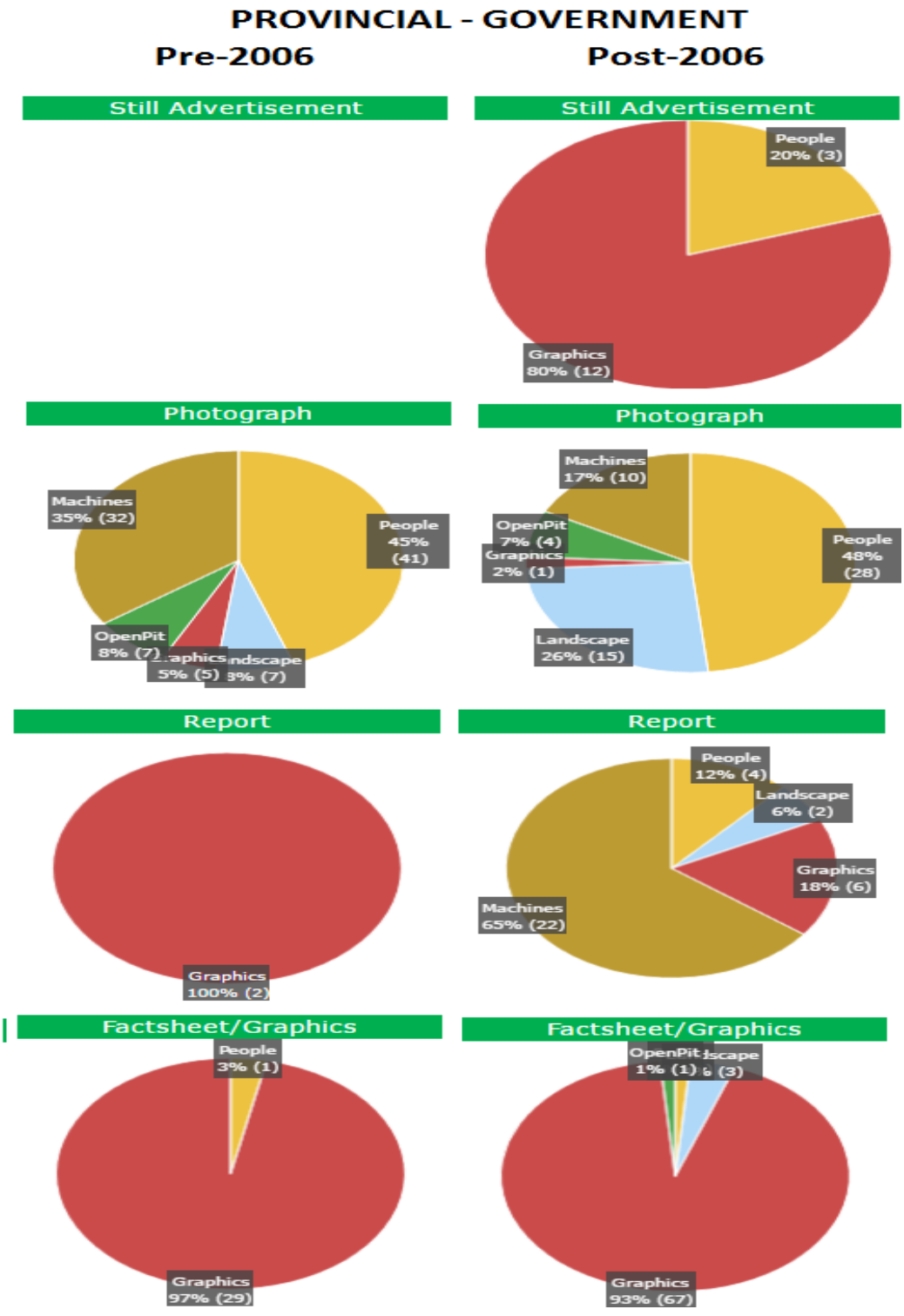


Figure 31. Stakeholder analysis: Provincial Government, pre-and post-2006

Figure 31 above shows the results of a comparative study of images produced by Provincial Government before and after 2006. It was observed from the documents that like Industry, Provincial government favours oil sands extraction, as it leads to economic benefits to the government in the form of taxes and revenues.

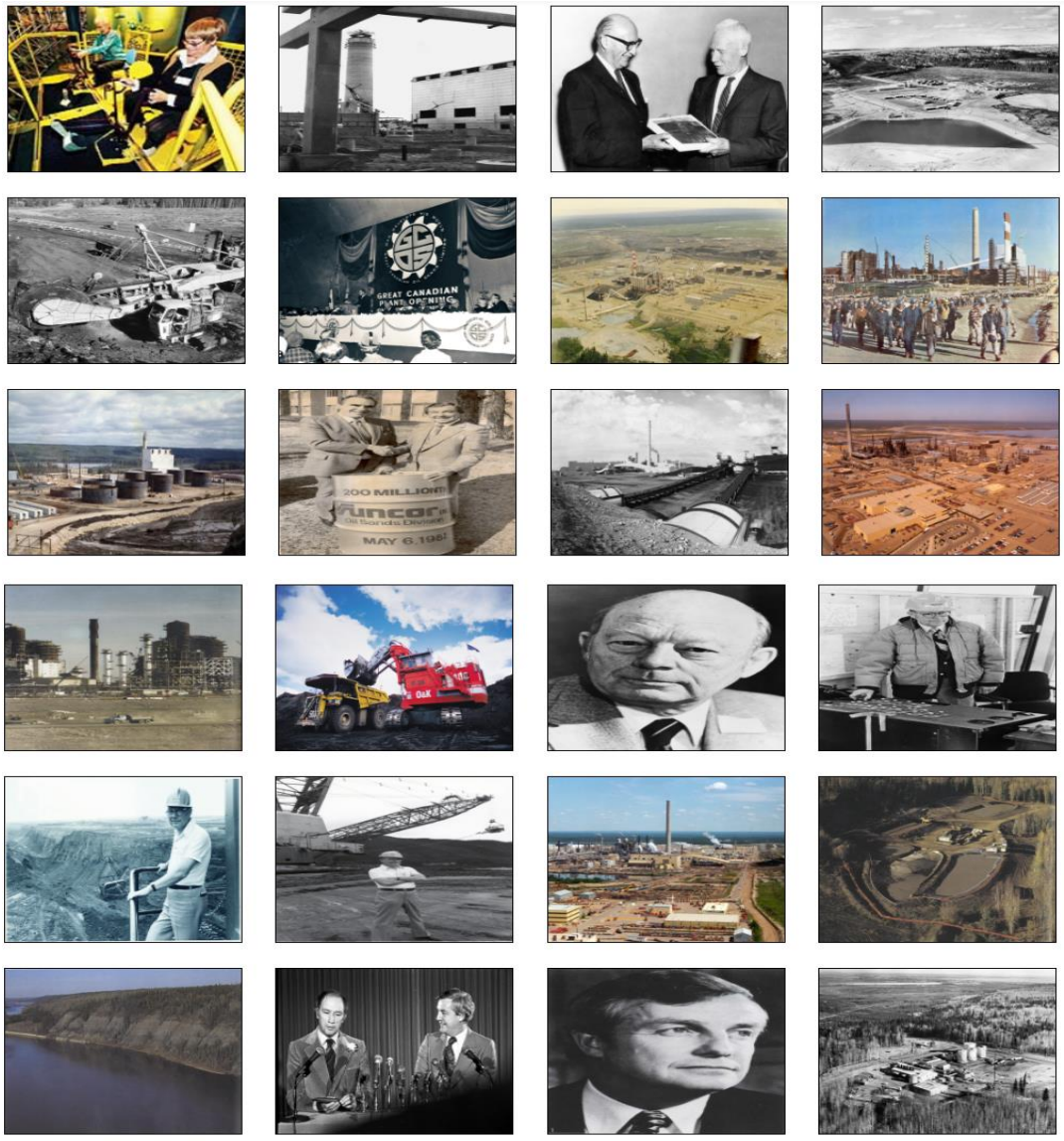


Figure 32. Photograph images in the Provincial Government stakeholder-type

Industry had only one document in the Photograph category before 2006. In contrast, Provincial Government commonly used Photographs in the same period. The reason for the latter phenomenon was that the images were retrieved from a digital archive maintained by the Alberta's provincial government, which deals with the history of the oil sands. In

Figure 32 above, nearly half of the images were archived black-and-white images that were produced by the Government of Alberta as part of its 'Alberta's Energy Resource Heritage' campaign. The two main content themes commonly found in these images were people (workers, eminent personalities in suits and ties) and oil sands infrastructure. The earliest recorded picture was from mid-1920s. After 2006, the use of People in photographs was reduced by 25%, while the use of Machines in images was reduced to one third in comparison to the earlier period. However, Landscape imagery showed an increase by a factor of two. Government and industries were facing rising opposition from civil society about the destruction of natural resources. Consequently, landscape images were used by government to show how well the oil companies are doing vis-à-vis returning the land to its natural state.

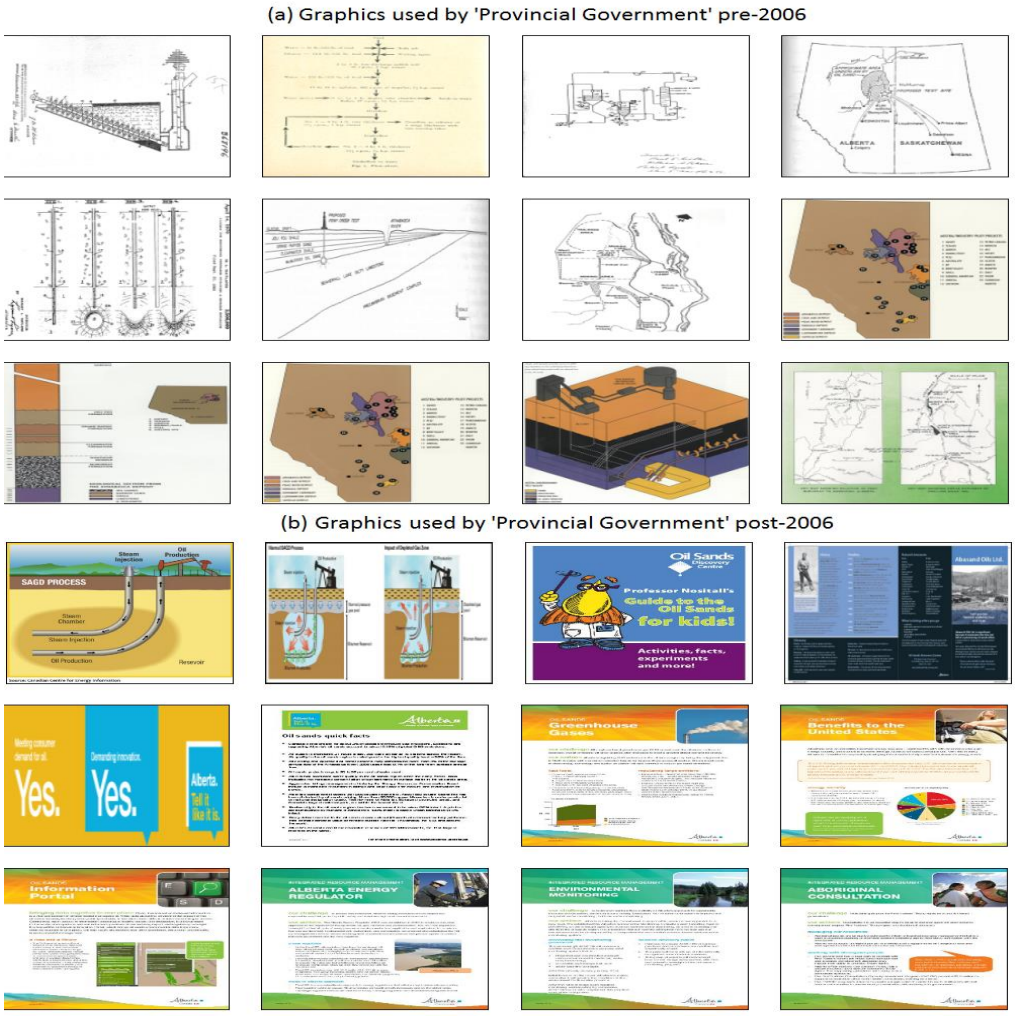


Figure 33. Factsheets/Graphics used by Provincial Government pre- and post-2006

In addition to Photographs, the Alberta's provincial government has used Factsheets as a tool for discussing tar sands extraction. The use of Graphics within the Factsheets has more than doubled. As shown in Figure 33 above, Graphics were primarily used pre-2006 for maps of oil sands reserves and to show different extraction techniques proposed by inventors. Thereafter the focus of Graphics shifted to showing the alternative to open-pit mining (i.e. SAGD) and depicting the corrective measures that industry is taking to preserve nature.

Only two Provincial Government reports (published in 1951 and 1982) are available in the database from the period before 2006. The period 2006 to 2015 saw a significant increase in the number of reports; the count is 34, of which 22 contain machines and infrastructure. All of these images were produced by Provincial Government as part of the "Alberta Oil Sands Industry – Quarterly Update" report. The Graphics, People, and Landscape categories respectively had shares of merely 18%, 12%, and 6%.

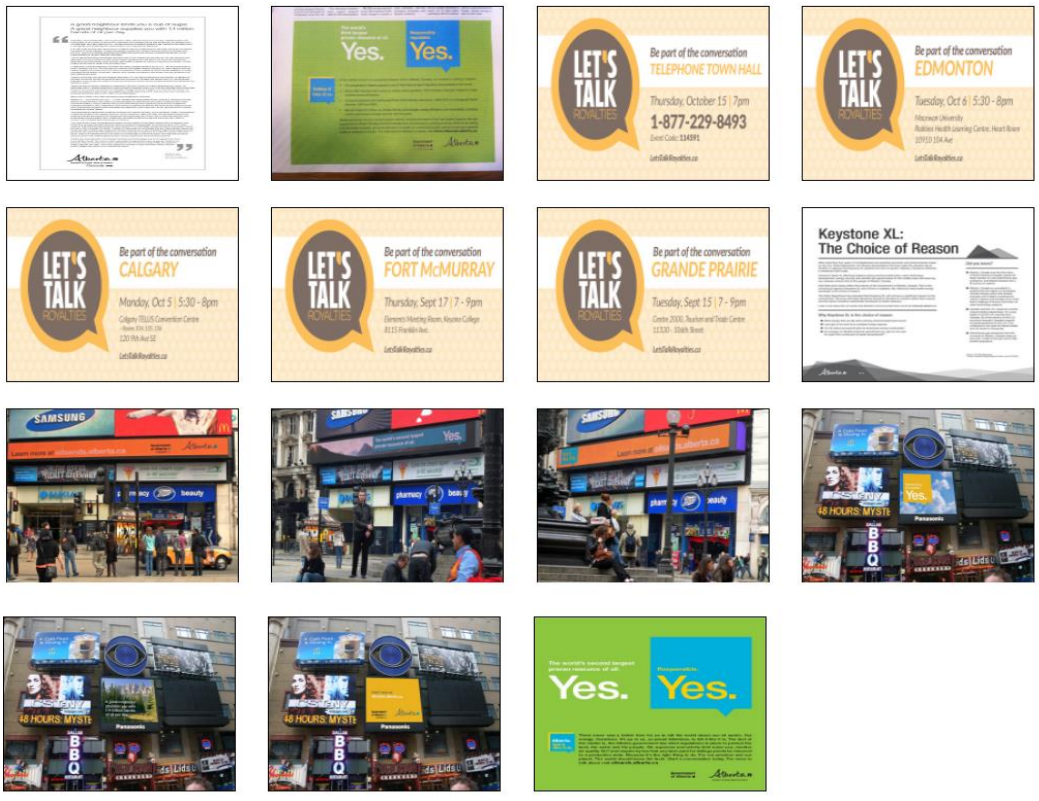


Figure 34. Still Advertisement images for the Provincial Government stakeholder-type after 2006

None of the Still Advertisements found during the data-gathering process were produced by the provincial government of Alberta before 2006; however, 15 advertisements were found in the later period, as shown in Figure 34 above. Out of the 15 available documents, 80% contain Graphics, while the remaining 20% include People. Upon further examining these images, we discovered that 5 out of 15 images were from the 'Let's Talk Royalties' campaign. The campaign talks about Albertans being the owners of the province's energy resources and their corresponding need to have a large share of the tar sands royalties [25]. As the images of digital displays showing Alberta's oil sands advertisements were actually taken on the streets of London (UK) and New York City, the presence of People in Still Advertisements was found to be irrelevant.[4]

4.3 Study of Different Stakeholder Categories, Post-2006

In the Mediatoil database, the earliest image captured from Industry is from the 1970s, while Provincial Government documents were available from 1925. However, documents from other stakeholder groups (Civil Society Pro-Oil Sands, Civil Society Anti-Oil Sands, Federal Government) started increasing around 2006. Our further discussion therefore focuses on a time series analysis of images from stakeholders after 2006. In Figure 35, 37, 38, 40, 43, 44 below, x-axis represents the year, and y-axis the represents the number of images available.

4.3.1 Stakeholder Time Series Analysis: Civil Society Pro-Oil Sands

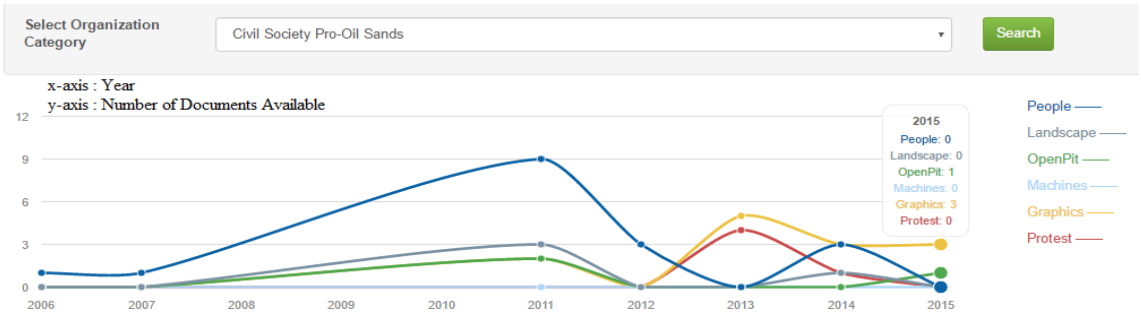


Figure 35. Stakeholder time series analysis: Civil Society Pro-Oil Sands

[4] The images where presence of humans was found to be irrelevant was from London, which may be accessed at: http://mediatoil.ca/Documents/Details/2728, http://mediatoil.ca/Documents/Details/2729; http://mediatoil.ca/Documents/Details/2730;

Only 42 images of civil society favouring oil sands extraction were found, and the Civil Society Pro-Oil Sands group contained just four stakeholders. Among these stakeholders, Ethical Oil produced 24 documents, Canada Action generated 15, and the Canada Center for Energy Information created 2; only 1 document was available from Canadian Energy Systems Analysis Research. The People curve played an active role until 2011. It subsequently ground to zero in 2013, increased by three in 2014, and finally levelled to zero again in 2015. Graphics show a significant importance after 2012 and were consistently used in pictures thereafter. Despite the low number of stakeholders in the Civil Society Pro-Oil Sands group, a few images belonging to the Protest content-type were identified. For instance, four pictures in 2013 and one image in 2014 were found in which people are demonstrating in favour of oil sands development, as shown in Figure 36 below.

Figure 36. Images of Protest found in Civil Society Pro-Oil Sands

4.3.2 Stakeholder Time Series Analysis: Civil Society Anti-Oil Sands

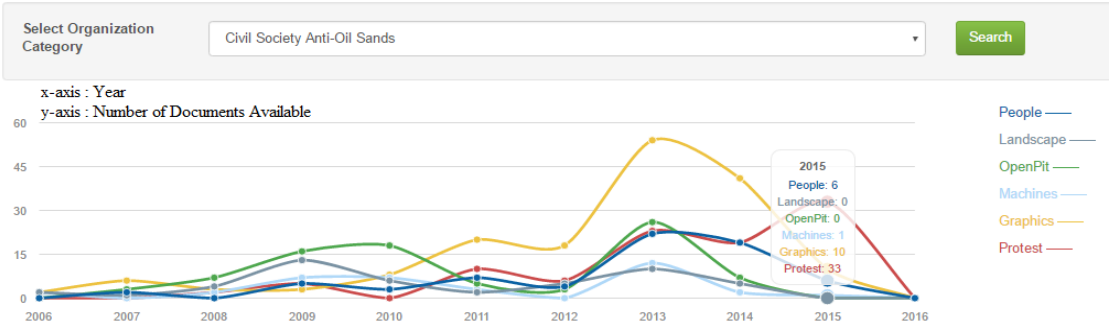


Figure 37. Stakeholder time series analysis: Civil Society Pro-Oil Sands

As indicated in Figure 37 above, 497 instances of civil society people were found, all against tar sands. Graphics combine text with image and make a picture more descriptive. They can be easily generated using image processing tools that are freely available. As Graphics are the cheapest source of advertising and visibility, Civil Society Anti-Oil Sands

thus widely utilized them; however, their usage showed a zigzag pattern until 2012. The Graphics curve escalated by 300% and reached its peak in 2013, with a maximum of 54 images as compared to 2012. It subsequently saw a decline over the remaining years. Another important image category is Protest. The curve showed a steady increase over time and in 2015 attained its highest point. The People class also saw a steady growth (with a few crests and troughs) until 2013, when its line achieved its highest value of 23. Apart from Protest, all of the categories attained their maximum height in 2013.

4.3.3 Stakeholder Time Series Analysis: Industry

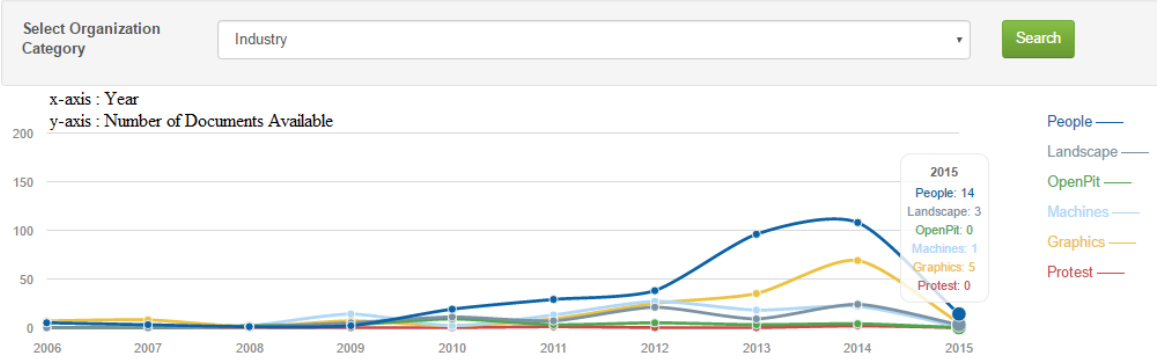


Figure 38. Stakeholder time series analysis: Industry

In Figure 38 above, the primary driving factor in Industry is People. A consistent increase was seen in the People category until 2014. As discussed in Section 4.1, Industry has wisely used its employees to show its great concern for preserving nature. Pictures of land reclamation as well as of land and water management were found. The use of Graphics was also significant, and it saw a consistent growth from 2010 to 2014. Industry stakeholders used very few pictures of Open-Pit mining, and those that they employed show accelerated dewatering. As discussed in Section 4.1 and illustrated previously in Figure 26, Suncor Energy also used Open-Pit images in its OSQAR blog to support constructive dialogues about the oil sands. Furthermore, Industry consistently utilized Machines to demonstrate ongoing extraction. The year 2014 was remarkable, as all categories apart from Protest surged upwards. In comparison to that year, only a few documents were available in 2015. Although a document may be active in the years following its creation, it is captured only once in the database (i.e. only in the year when it is produced). It is possible that documents

produced prior to 2015 were still actively being used by Industry in 2015. This may be one explanation for why all categories dwindled and levelled nearly to zero in 2015, with the exception of People (which still had 14 instances). Industries also carried out campaigns to support oil sands development. As shown in Figure 39 below, three instances of Protest imagery from Industry were found: one from Suncor Energy (2011) and two from TransCanada (both from 2014.

Figure 39. Protest images in the Industry stakeholder-type: Protest image from Suncor Energy in 2011 (left), Protest images from TransCanada in 2014 (middle, right)

4.3.4 Stakeholder Time Series Analysis: Provincial Government

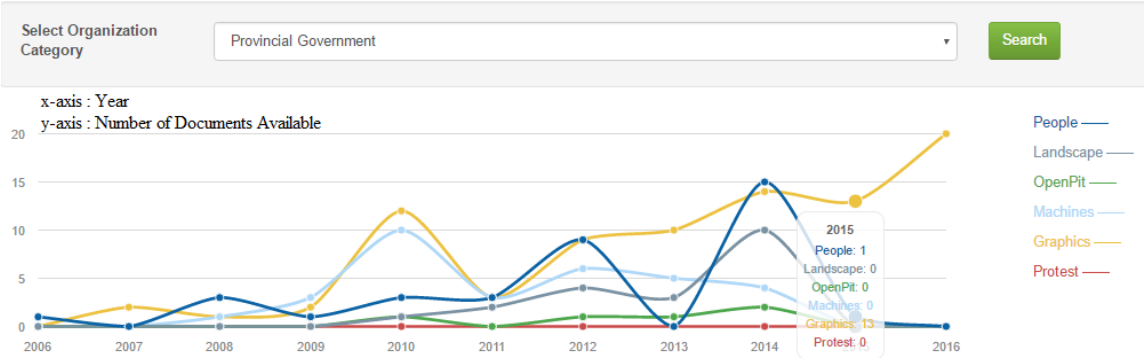


Figure 40. Stakeholder time series analysis: Provincial Government

Alberta's Provincial Government has been an active player in the oil sands imagery. Specifically, the graph in Figure 40 indicates that Graphics and Machines pictures were most widely used in 2010. Although only 12 and 10 images respectively contained Graphics and Machines, these numbers were the highest among all content-types. In the 2011, the use of both kinds of images declined and levelled equally to three. Furthermore, Graphics saw a steady growth while the use of Machines showed a decrease over the following years. People and Landscape images attained their peak levels in 2014, with 15 and 14 images, respectively. Figure 41 below shows that the Government of Alberta

employed Graphics in its ‘Alberta. Tell it like it is.’ campaign to highlight facts regarding responsible oil sands development. As depicted in Figure 42 below, Graphics also witness an interesting shift from open-pit mining to SAGD after 2006. In contrast, the Protest class was not found in any of the images.

Figure 41. Images from the Government of Alberta’s 2010 ‘Alberta. Tell it like it is’ campaign

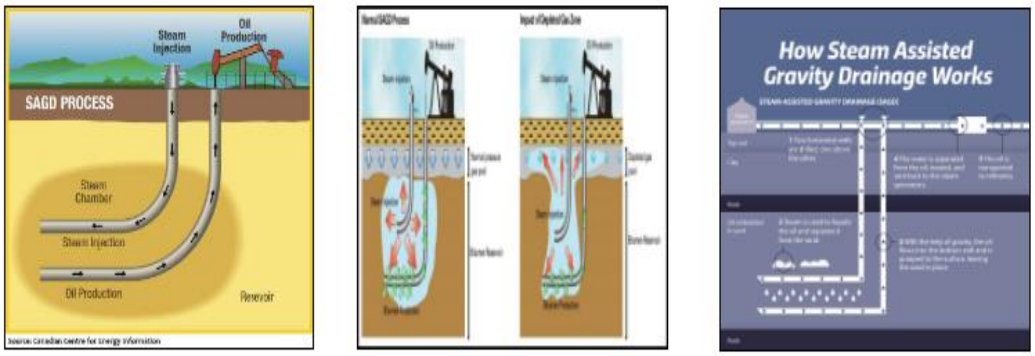


Figure 42. Images from the Government of Alberta showing he SAGD process in 2010 (left, middle) and 2015 (right)

4.3.5 Stakeholder Time Series Analysis: Federal Government

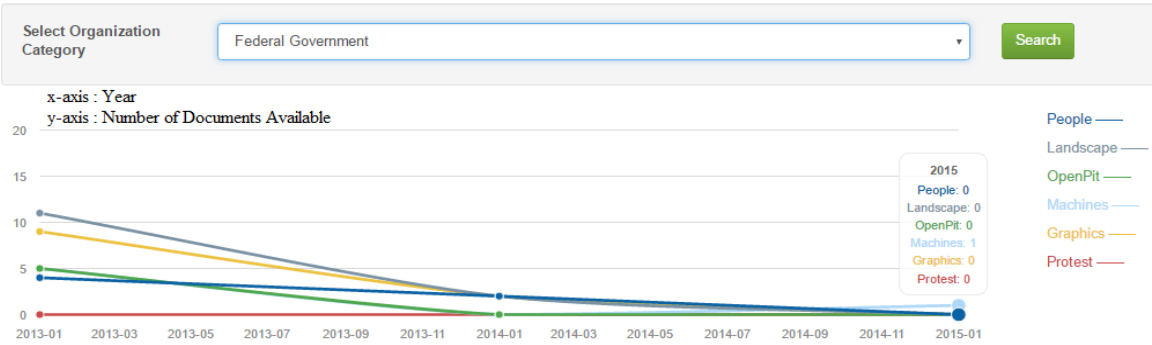


Figure 43. Stakeholder time series analysis: Federal Government

Although we were able to collect only 41 images from the Federal Government, the results suggest that these documents favour tar sands. The claim stems from the fact that 40 of these 41 pictures were from the 'OPEC Lobbyist of the Year Awards' campaign [21], which was hosted by EthicalOil.org – a civil society organization that favours ethical oil being produced inter alia from oil sands [21]. The remaining image was from the 'Boycott Chiquita' campaign. EthicalOil.Org launched this campaign against Chiquita Brands International Inc., which is "*an American producer and distributor of bananas and other produce*" [108]. In this campaign, it requested "*all Canadians to boycott Chiquita brands until they reverse their boycott on Canada's ethical oil!*" [22]. Very few documents from Landscape (11), Graphics (9), Open-Pit (5), and People (4) were found in 2013, and later years witnessed further dips in all of these categories. Similar to the situation in Provincial Government, the Protest class was again not part of any of the images.

4.3.6 Comparison of the Three Main Players: Suncor Energy, the Government of Alberta, and Environmental Defence

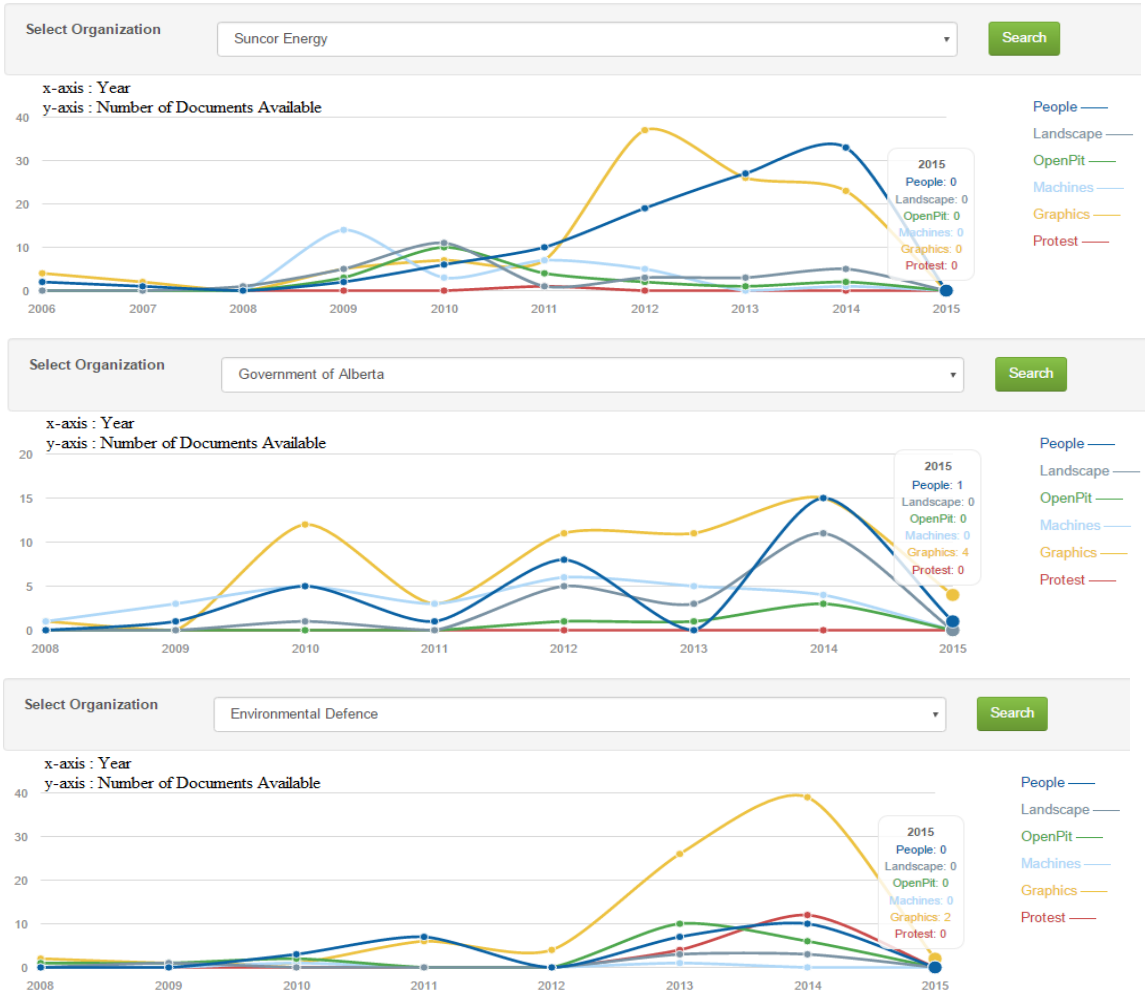


Figure 44. A comparison of imagery used by the three main players in the oil sands debate: Suncor Energy, the Government of Alberta, and Environmental Defence

Figure 44 above shows a comparison of Suncor Energy, the Government of Alberta, and Environmental Defence. Based on the number of documents available, these entities are leaders in Industry, Government, and Civil Society, respectively. Although Civil Society Anti-Oil Sands stakeholders may have been active before 2006, we were only able to collect one image each for 2004 and 2005. Considering these years insignificant on the basis of their low document counts, we decided to begin our comparison with 2006. The evaluation thus includes pictures from 2006 to mid-2016 (when data collection stopped). From the graphs, it was clear that Suncor Energy was only actively giving its views on the tar sands in the years from 2006 to 2008; however, we were not able to collect any

documents for the other two organizations for 2006 and 2007. In 2008, the Government of Alberta focused on the controversial tar sands by utilizing Machine/Infrastructure pictures as the instrument for describing its views. Suncor Energy proceeded in the same way, although in addition to Machines they also promoted their thoughts using the Open-Pit and People classes. In contrast, as Environmental Defence was in its inception phase, they have only three documents in that year.

The three entities experienced divergent patterns from 2009 to 2012. Suncor Energy showed a variation during this period by focusing less on Machine pictures and more on the People, Open-Pit, and Graphic groups. In comparison, the Government of Alberta showed trough and crest patterns for Graphics, People, and Open-Pit levels and Environmental Defence highlighted People and Graphic sections. In the following two years, all three entities actively portrayed their views on the tar sands. Environmental Defence, which had been partially inactive in the previous few years, showed a keen interest by concentrating on Graphics. The Government of Alberta also chose Graphics along with People and Landscape pictures as tools for outlining its opinion; Suncor Energy also primarily centralized on Graphics and People. The year 2015 was marked as a year of low interest, as the three top participants appeared to witness a steady degradation in their views on bituminous sands. This may be because we do not have sufficient data for 2015 in the Mediatoil database.

4.4 Discussion

The aim of this chapter was to differentiate the images used by various stakeholders based on the images' contents. Six major content-types were identified and defined (People, Machines, Graphics, Protest, Open-Pit, and Landscape). We recognized that the Civil Society Anti-Oil Sands category came into existence in 2004, but according to the data available it became more active from 2006 onwards. In the earlier period, only Industry and Provincial Government documents were available. In section 4.1, we therefore differentiated the imagery utilized by Industry based on whether it was pre- or post-2006. Before 2006, Industry published mostly annual reports, and documents mainly contained computer-generated Graphics. Photographs, Still Advertisements, and Factsheets all saw tremendous growth after 2006. An emphasis was placed on using humans in the documents

to highlight the economic benefits of tar sands and show that the oil sands are being responsibly developed in a way that minimizes negative environmental impacts. A similar evaluation was conducted for the Provincial Government stakeholder-type. Before 2006, Photographs mainly contained People and Machines, while Factsheet largely included Graphics. Most of the images were black and white, and they included renowned personalities in suits and ties, machines, and oil sands extraction plans. In the later period, a few Landscape images were used to demonstrate how well the oil companies are doing on returning the land to its natural state; some Graphics were also used to show in situ oil sands extraction.

A time-series analysis of different stakeholders was conducted in Sections 4.3.1 through 4.3.5. Graphics were commonly found in all stakeholder images, as it is a cheap source of advertisement. Graphics and Protest images were mostly found in the Civil Society Anti-Oil Sands stakeholders group, whereas Graphics and People were mainly found in Industry images. The years 2013 and 2014 were remarkable, in that all the stakeholders produced the maximum number of documents for the period considered. Industry and Government groups were in favour of the oil sands and producing documents that could impact life, society, culture, or the environment. Their focus was on showing oil as an essential commodity in every aspect of our existence. Significantly, a few Protest pictures from Industry and Civil Society Pro-Oil Sands were also found.

4.5 Summary

In this chapter, we explored the imagery used by stakeholder groups over the years. Industry and Provincial Government were two primary stakeholder groups that were active before 2006. Therefore, detailed analyses of these groups were conducted for pre-a and post-2006 timeframes. A complete study of images used by different stakeholder groups after 2006 was also undertaken. The analysis of pictures required a CBIR for image retrieval.

In the next chapter we discuss the Mediatoil-IR CBIR system, which not only acts as a medium to retrieve archived images but also serves as a framework that is open to all researchers studying oil sands.

Chapter 5

The Mediatoil-IR

In the previous chapter, we explored the different types of pictures used by the distinct stakeholders, based on the images' contents. Deciding the content-type was a crucial task and required utilizing feature extraction and machine learning, which are covered in this chapter. The Mediatoil-IR is built on the intuition that if we can predict the class of the query image, then a smaller dictionary of similar images can be searched for image retrieval.

Figure 45 below contains the framework of the Mediatoil-IR. The diagram is a modification of the structure in [55] that uses a common dictionary for all classes of images, image searching, and retrieval. In contrast to the concept in [55], Mediatoil-IR builds individual search dictionaries for different categories of pictures. The rationale behind using separate search collections is that retrieval accuracy can be improved if we look at images in small collections of related documents [39]. For instance, we have created dictionaries for each of the six categories (People, Protest, Landscape, Open-Pit, Graphics, and Industry). We also used supervised machine learning algorithms to train models that classified images into given categories. Finally, a document retrieval process was followed once the machine learning classifiers had determined a query picture's class.

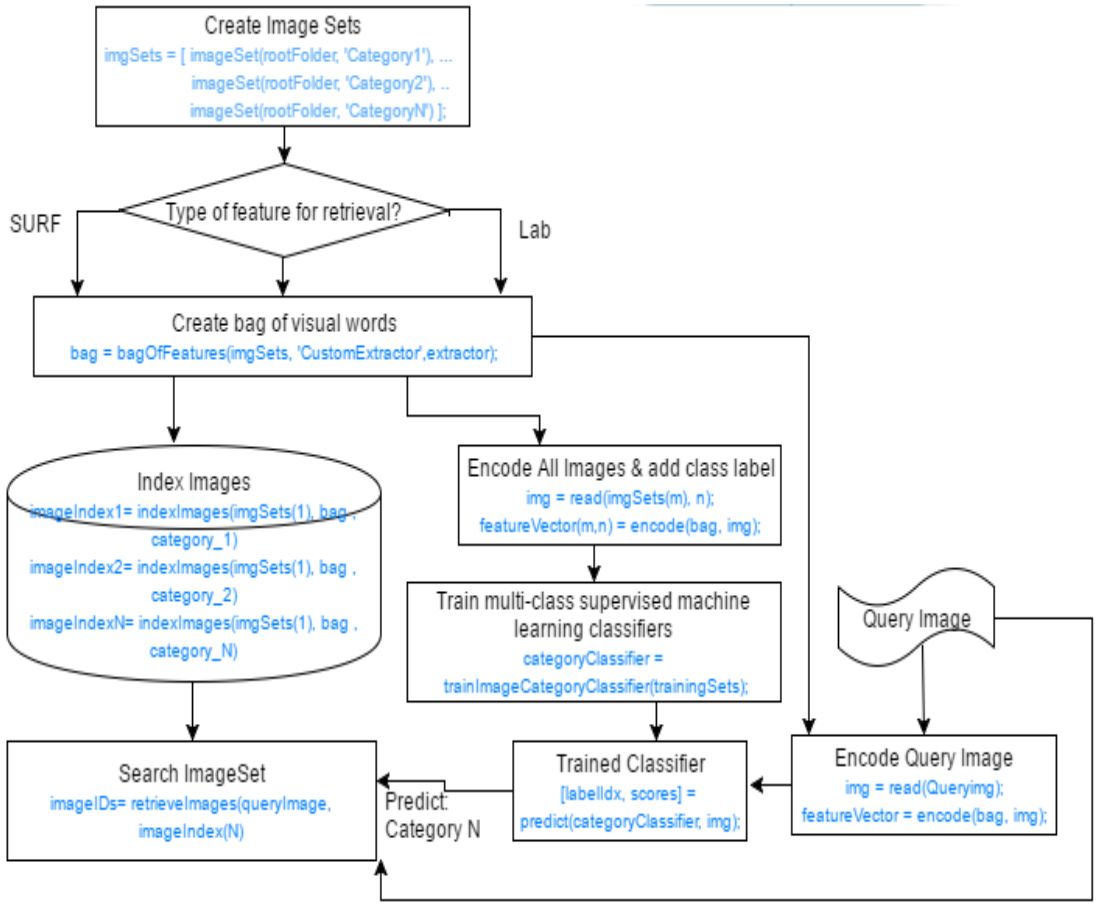


Figure 45. Framework of the Mediatoil-IR [55]

The Mediatoil-IR framework has two phases, namely training and image retrieval.

5.1 The Training Process

In the below pseudocode for training, IM is the complete image set, where IM_1 = People, IM_2 = Protest, IM_3 = Machines, IM_4 = Open-Pit, IM_5 = Landscape, and IM_6 = Graphics images. In the first step, the features from all of the images were combined to form a bag of visual words (BOW). The ExtractSURF or ExtractLab function then extracted the respective features.

The second step entailed performing quantization to reduce vocabulary size to given size K; otherwise, a vocabulary may contain millions of features. A large vocabulary size not only leads to a large feature size for an image; it also increases the search dictionary size and makes searches slow.

Algorithm- Training Mediatoil-IR

Input: IM is the set of all images organized in different sub-folders based on their content category, and IM ⊇ {IM_1 …. IM_N}.
FT ∈ {"SURF", "Lab"}is the feature type to be used.
CL ∈ {"Linear/Quadratic SVM", "DT", "K-NN", "Bagging", "Boosting"} is the base classifier.
BOW is a combined set of all features from all images; initially, it is empty.
K is the maximum required vocabulary size.
COW is the cluster of visual words; initially, it is empty.

1. **for all** images i in IM **do**
    a. **if** (FT== "SURF"), **then**
        FS = extractSURF(i)
    else
        FS = extractLab(i)
    end if
    b. BOW = append (BOW, FS)

end for

2. **if** length(BOW) > K, **then**
    COW = K-means (BOW, K)

else
    COW = K-means (BOW, length(BOW))

end if

3. **for all** images p in IM **do**
    a. **if** (FT== "SURF"), **then**
        FS = ExtractSURF(p)
    else
        FS= ExtractLab(p)
    end if
    b. Initialize hist[p, 1 : K+1] = histogram of visual words of length K+1
    c. **for all** f in FS **do**
        i. c = evalCluster(COW, f)
        ii. hist[p, c] = hist[p, c] + 1
    end for
    d. hist[p, K+1] = class of image

end for

4. TM_1, TM_2, TM_3 = model induced by CL on training data hist
5. **for** z = 1 : 6 **do**
    a. for all visual-word w in COW **do**
        i. for y = 1 : count(hist) **do**
            if (hist[y, w] != 0)
                SI_z[w] = append(SI_z[w] , (y, hist[y, w]))
        end for
    end for

end for

Output:

1. TM_1, TM_2, TM_3, are three training models build using CL
2. SI_1, SI_2, SI_3, SI_4, SI_5, SI_6, are six dictionaries (one for each category of image)
3. COW is clusters of visual words

Figure 46. Pseudo code for training Mediatoil-IR

The K-means algorithm was used to perform quantization, which is undertaken to cluster all given features into K clusters such that the clusters are mutually exclusive. The K-means algorithm "*divides M points in N dimensions into K clusters so that the within-cluster sum of squares is minimized*" [30]. "*The K-means algorithm is a widely used algorithm that has been shown to be efficient and scalable, and fast converge when dealing with larger data sets*" [101]. As defined in [101], the main limitation of K-means is that the number of clusters needs to be determined in advance. In our work, the number of clusters, K (10000), was chosen carefully and after experimentation. The outcome of quantization is COW, which represents the clusters of visual words, where the mean of each cluster is identified as a visual word.

In the next step, we binned each extracted feature in the appropriate cluster and subsequently incremented the bin count for this cluster by 1. The final hist[p], histogram became the final descriptor of a picture 'p', where each histogram had a length of S+1. The first S features corresponded to the bin count of each feature, whereas the last feature (S+1) denoted the class of the image (People, Protest, Landscape, Open-Pit, Machines, Graphics).

In the fourth step, three classifying models (TM_1, TM_2, TM_3) were prepared using CL as the base model. TM_1 categorizes People vs. Non-People, TM_2 classifies People vs. Protest, and TM_3 classifies from the rest four categories (Landscape, Open-Pit, Machines, Graphics). These three models were learned to follow a multi-step process for identifying an image's content-type. Finally, six dictionaries were created (one for each category). The key in the dictionary is a visual word, and the value is a pair (image index, count) that reflects the index of the image that contains the given visual word and the number of times that word appears in the given picture.

We next discuss these steps in detail.

5.1.1 Creating Image Sets

The first step in the training process is to separate all images into different folders based on the content-type to which they belong. Our initial task was therefore to assign a class label manually to each image using the six given content-types. Here it should be recalled

that as the pictures were divided into six class-types, there were six image sets. It is desirable to have a large number of images in each image set to improve the accuracy of retrieval. The task of separating images into different folders was done with the help of PI from the communications department.

5.1.2 Selecting Types of Features for Retrieval

As noted in the literature review in Chapter 2, we concluded that the Lab color space and SURF features methods are most applicable for the MediaToil-IR system. This is due to the fact that the LAB space has a wider range than other color spaces [105] and is device independent, i.e. colors are defined independently of their origin or the device they are displayed on [105]. The SURF feature method was also found to be more robust than other available techniques, such as Harris, SIFT, FAST, BRISK, and MSER [31, 38, 64, 76].

Each LAB feature is a five-dimensional feature that contains values for L, a, b from LAB color itself and x, y from that feature's spatial location. During the process of identifying LAB color features, the first step entails converting the training images from RGB color space to LAB color space. This is done as the latter enables the visual differences between colors to be quantified quickly. Visually similar colors in the LAB color space have small differences in their LAB values. The next step is to perform L2 normalization on the LAB features to discard the delicate features. Only the sharp features will sustain after normalization. Thirdly, spatial augmentation is performed on normalized features, where [x, y] location of an element is embedded with the feature vector [54]. Two images have similar color features if their color and spatial distribution of color are similar, which makes this technique a better choice than other color schemes.

As indicated previously, the SURF detector is much faster and more robust than other non-binary detectors [3]. Furthermore, as noted in chapter 2, the SURF descriptor describes the features in such a way that they are rotation and scale invariant. Each image produces around N (~2000-4000) features, where each feature length is 64 words [52].

Once the feature set to be used has been selected, the next step is to combine all of the features into one set and prepare a BOW, as explained below.

5.1.3 Creating a Bag of Visual Words

A standard technique for implementing a CBIR system is the bag of features [73, 83]. The method bag of features, also known as bag of visual words (BOW) is adapted from the worlds of document retrieval and natural language processing to image retrieval [54]. Rather than using actual words as in document retrieval, the BOW uses image features like the visual words that describe an image [54]. Instead of creating a set of features for each image, BOW allows one global vector to be created per image. This one global descriptor can be either compared for similarity matching at runtime or be indexed.

Figure 47. Creating a BOW [53]

Figure 47 above depicts the process of building a BOW. The foremost step is to identify feature point locations, either by creating a grid or relying on a particular detector that has an independent feature point detection algorithm. The LAB color feature uses a grid approach, whereas SURF uses its internal procedure for identifying feature points. In the grid method, each grid cell is defined as one feature point.

The next step is to create a local descriptor for each feature point. In the grid technique, the average LAB color is computed within 16 X 16 pixel blocks. The average color for each feature point is expressed as a 3D value. The point's x- and y-location is also added to the feature descriptor, thus creating a five-dimensional (5D) descriptor. In contrast to the 5D descriptor, the length of each SURF feature is 64 words. Recall that the process of SURF detector and descriptor extraction is covered in Chapter 2.

The third stage in creating a BOW is to successively append all of the local feature descriptors from all of the images in the given six classes to form a large bag of all feature descriptors. Following concatenation, quantization is performed to reduce the number of features [52]. Quantization is necessary as a BOW may contain millions of features and due to large dimensionality, image retrieval can be slow. The number of features can be

reduce by clustering the same features together. For instance, as discussed in Section 5.1.2, each image generates N (~2000-4000) descriptors; comparing these features at run time in a brute force manner with all of the available images is not possible in real time. In contrast, BOW creates one global descriptor for an entire picture, which is a fast and efficient way to make comparisons with other images descriptors in real time. In Section 5.1 it was noted that the K-means algorithm is applied to cluster all of the features in k-clusters and that the cluster centers represent the chosen visual words. As similar features are grouped together, this clustering may reduce the millions of features to a few thousand. The global feature vector for each image will consequently have few features, which can make the retrieval faster.

Different image categories may have an uneven number of instances, which leads to the feature detector algorithm producing a different number of visual words for each category of pictures. If all of the features from all of the classes are used for clustering, the result will be more keywords from the majority classes and fewer keywords from the minority classes. This enables the global descriptor, which is encoded from BOW, to have more visual words from the majority classes. Furthermore, it will be biased towards the classes that contain greater numbers of visual words.

A global descriptor should therefore contain an equal number of keywords from all of the categories, so that it is not biased towards any particular class. As a result, N number of strongest features from each category was equally chosen to be clustered. For example, if the Open-Pit category has a minimum number of instances, then features of the Open-Pit type extracted by a feature detector algorithm will also be minimal amongst other categories. Assuming the detector extracted N features for Open-Pit, the top N strongest features were taken from each other class before clustering, which yields equally sized clusters. The strongest features were decided on the basis of the score assigned to each keypoint by the feature point detector [52]. The global descriptor subsequently encapsulated an equal number of visual words from each class. A vocabulary size of X was chosen to enable the efficient incorporation of the features of all images. After inspection, X was set to 10000, which resulted in 10000 visual words.

5.1.4 Encoding Images and Preparing Image Indexes

The BOW provides a terse encoding scheme to identify a large set of images using a sparse set of visual word histograms [52]. The encoding process yields a global descriptor for a picture. Instead of producing variable length features for each image, it provides a single descriptor of length N, where N is the size of a bag. The encoding process is covered in Section 5.2.1.

These histograms act as a global feature and can be stored for later matching and retrieving similar images. Finding similarities at run time by matching a query image's histogram with all other histograms is a time-consuming process and is feasible only if the dataset is small. It can lead to long wait times for large datasets, which is not desirable in real-time systems. The features are therefore indexed in a dictionary.

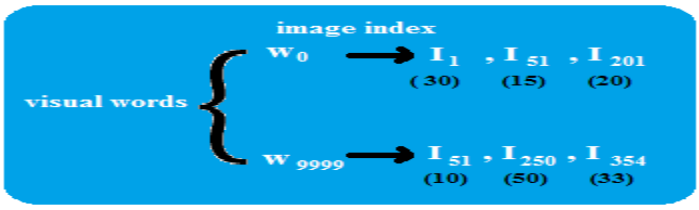


Figure 48. An example of an image dictionary

Figure 48 above describes an image dictionary. Creating a dictionary may assist in fast retrieval, unlike brute force matching. For each visual word, the dictionary contains a list of images that includes that word and the number of times that word appeared in the given picture. For example, visual word W_0 appeared 30 times in first image, 15 times in picture number 51, and 20 times in photograph 201.

Once the BOW is created and each image is encoded in a visual word histogram, the entire image set can be indexed for search [54]. In general, one search index is created for all of the pictures. The uniqueness of our approach here is that instead of creating one single dictionary for all images, we created six different dictionaries, such that each category has its separate image list. This means that results are retrieved by searching images from a single dictionary and are more relevant.

5.1.5 Training Supervised Multi-Class Classifier

In the previous section, we postulated that results would potentially be more accurate if images were searched from one specific dictionary. Before retrieving results, we need to identify the image collection to which the query image belongs. This identification is achieved with the help of supervised machine learning. A single histogram for one image, generated by an encoding method, corresponds to a feature vector of length N, where N was set to 10,000 (by inspection). The histograms (along with their category label) are given to the supervised machine learning algorithms for training. Different supervised machine learning algorithms were covered in Section 2.4. These algorithms behave differently on various datasets. To determine which algorithm will be most suitable for the Mediatoil database, we experimented with linear SVM, quadratic SVM, weighted K-NN, DTs, bagging, and boosting. The basic approach was to learn a multi-class classifier, but the classes were overlapping. As a result, we divided the task into three sub-problems, with the aim of differentiating image categories as much as possible. Labeling an image as belonging to one of the six classes therefore became a multi-step process, as outlined below.

Our first classification task was to train classifiers to learn models for the binary task focussing on **People vs. Non-People**. As people may appear in either separate photos or photos in any other category (such as Landscape), it was mandatory to split the People class from the other five categories. If a picture belongs to a Non-People category, it often does not belong to Protest, as protest pictures usually involve humans. As a result, machine learning-based classifiers were learned, where People and Protest images were combined as People and remaining four classes were combined as Non-People.

The second classification task contrasted **People vs. Protest**. It follows that if a picture is identified as containing a human, it may also be a part of a protest. People and Protest images were thus classified separately by a binary classifier.

Finally, we explored a multi-class learning task, i.e. if an image belongs to a Non-People category, then it fits either the **Machines, Open-Pit, Landscape or Graphics** class. A multi-class classifier is required to classify images into one of the four types.

5.2 The Image-Retrieval Process

Algorithm- Image-Retrieval Mediatoil-IR

Input: Q is the query image you want to search, and look its similar images.
TM_1, TM_2, TM_3, are three training models build using CL in the training step
SI_1, SI_2, SI_3, SI_4, SI_5, SI_6, are six dictionaries (one for each category of image)
COW is clusters of visual words generated in training
FT is feature type used in training
TC is set of target classes
N is number of return results required

1. **if** (FT== "SURF"), **then**
   FS = ExtractSURF(Q)
   else
   FS= ExtractLab(Q)
   end if
2. S=size(COW)
3. Initialize hist[Q, 1 : S+1] = histogram of visual words of length S+1
4. **for all** f in FS **do**
   a. c = evalCluster(COW, f)
   b. hist[Q, c] = hist[Q, c] + 1
   end for
5. hist[Q, S+1] = ?
6. C = predict(TM_1, hist[Q])
7. **if (**C == TC[1]), **then**
   a. C = predict(TM_2, hist[Q])
   b. **if (**C == TC[1]), **then**
   ItS = SI_1
   else
   ItS = SI_2
   end if
   else
   a. C = predict(TM_3, hist[Q])
   b. **Case based on C**
   Case TC[3]
   ItS = SI_3
   Case TC[4])
   ItS = SI_4
   Case TC[5]
   ItS = SI_5
   Default
   ItS = SI_6
   end Case
   end if
8. [Idx]= retrieve top N results from ItS image dictionary using hist[Q] as feature-set.

Output:
1. [Idx]- Image ID's of top N images.

Figure 49. Pseudocode for query retrieval Mediatoil-IR

The retrieval algorithm takes as input inter alia the query image, identifies the category of the picture, and returns the index of top N similar results from the designated image search dictionary. Figure 49 above contains the pseudocode for image retrieval.

In this pseudocode, TC represents the set of target classes {People, Protest, Machines, Open-Pit, Landscape, and Graphics}. The first four steps of the process entail extracting features from an image and encoding the image into a feature vector. The fifth step just indicates that the TC is unknown. The next stage involves predicting the TC of a query image encoded in hist[q].

Classification of a picture into one of the six categories requires a three-step process: TM_1 categorizes People vs. Non-People, TM_2 classifies People vs. Protest, and TM_3 labels from the remaining four categories (Landscape, Open-Pit, Machines, Graphics). The reason for using a three-step process is explained in Section 5.1.5. Once the class of image content is known, a particular dictionary can be searched to retrieve top N results; this increases the overall accuracy and speed in comparison to searching for all of the categories in a common dictionary.

All of the steps followed in the image retrieval process are elaborated in the following discussion.

5.2.1 Encoding a Query Image

Section 5.1.4 explained that preparing a BOW also generates an encoding method that creates a histogram of the visual word occurrences contained in an input image [54]. The query image is then fed through an encoding process and a feature vector, also known as visual word histogram, of size N (10000) is generated.

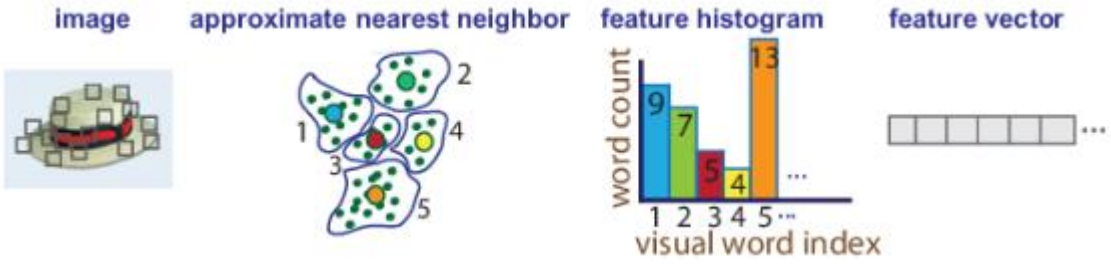


Figure 50. Extracting a global feature vector using BOW [53]

Figure 50 above illustrates the process of converting an image into a feature vector. The ***encoding process*** for a single image encompasses extracting its features one by one, approximating each feature to the nearest visual word, binning it into the appropriate bin, and incrementing the count of that bin by one. The result is a visual word histogram of length N for each image. This histogram shows the word count for each word in the query image. As each image has an individual histogram, this histogram can be used for similarity matching and image retrieval. The histogram can therefore be represented in the form of a vector, where each visual word accounts for a feature and calculated histogram count specifies a feature value. This feature vector enables compact storage with each image stored as having a similar length feature.

5.2.2 Predicting Image Category from the Multi-Class Classifier

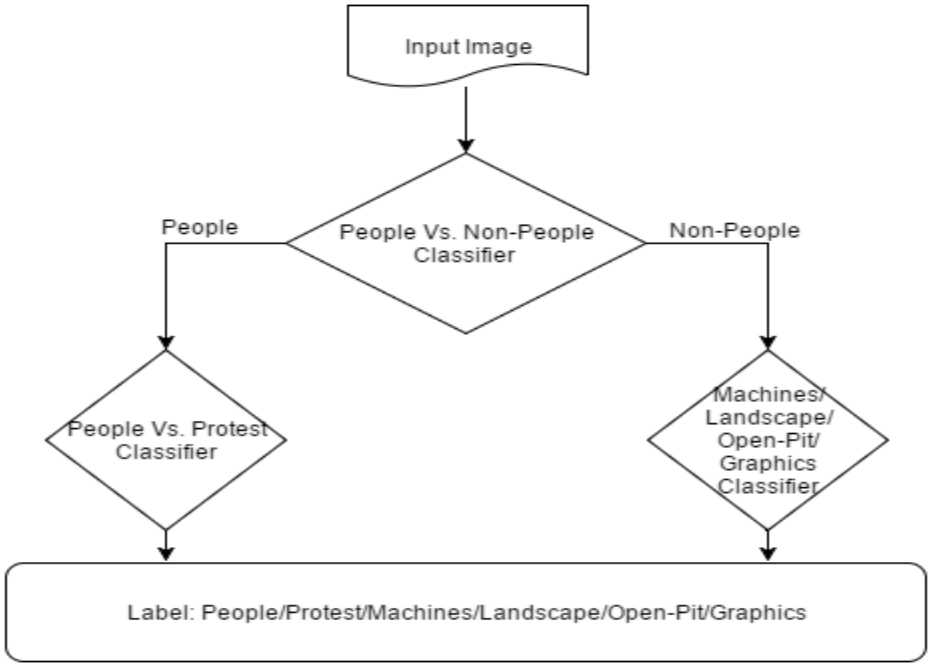


Figure 51. The multi-step process of classifying an image

Figure 51 above shows that once converted into a feature vector, the query image is classified by the best-learned classifier (linear SVM, quadratic SVM, weighted K-NN, DT, bagging, boosting) in one of the six categories. There are three classification models, each of which handles a subset of classes. The efficiency of the retrieval process depends on the accuracy of each trained classifier. In general, higher accuracy of the trained classifier will lead to more efficient retrieval results.

5.2.3 Retrieving Images from the Respective Search Dictionaries

Based on the class predicted by the best-learned classifier in Section 5.2.2, the search is conducted within a particular image collection. The results are image IDs and similarity scores.

For instance, Figure 52 below illustrates the process of seeking similar pictures for a query image once its category has been determined. The given picture is encoded in a visual word histogram and provided to the search dictionary that is predicted by a classifier in the previous step. It should be recalled that the search dictionary contains the vocabulary of all of the visual words and indexes to images for each word. The search process yields the image indexes for all of the pictures that match the query image's visual words, with scores sorted from best to worst [54]. By default, the maximum number of retrieved images is set to 20; however, it can be changed to any positive number or to infinity (for finding all matching pictures).

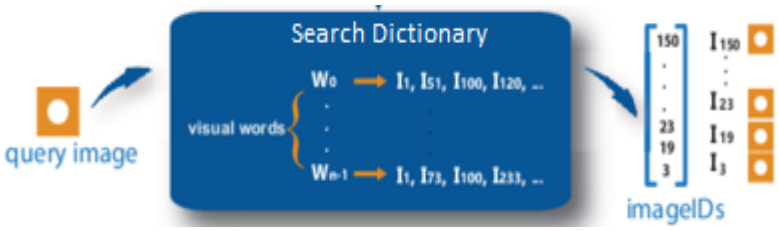


Figure 52. Searching similar images in a search dictionary

5.3 Summary

In this chapter, we discussed the process of creating an image retrieval system for oil sands images. In brief, this process comprises two steps: training and retrieval. The training step includes grouping images into respective classes, extracting features, preparing BOWs, creating a histogram for each photograph, using this histogram for developing a feature vector, and ultimately giving this vector (along with its class label) to a supervised machine learning-based algorithm for training models. The retrieval process involves encoding a query image into a global feature vector, predicting its type with the help of learned machine learning models, and retrieving similar images from the respective search dictionary. The next chapter discusses our experimental setup and evaluation.

Chapter 6

Experimental Evaluation

Measuring the performance of different machine learning algorithms is crucial for choosing classifiers to accurately typify the imagery in the Mediatoil-IR systems. In this chapter, we discuss the experimental setup, describe various evaluation measures, and finally evaluate different algorithms on the selected feature spaces. Our experiments include the performance of the LAB color and SURF features methods on several machine learning algorithms (linear SVM, quadratic SVM, weighted K-NN, DTs, bagging trees, boosted trees).

6.1 Experimental Setup

The experiments are conducted in Matlab 2016b on a 64-bit architecture running Windows7 with 16 GB RAM. The parameter settings for different algorithms are discussed in this section. The DT used classification and regression trees (CART) [8] algorithm which uses Gini's diversity index as the split method, as it is faster than other purity measures given that it does not include log computation; to keep the tree simple to interpret, the maximum number of splits at each level was set at four. In weighted K-NN, the Euclidean distance metric with the number of neighbours equal to 10 was chosen and set after experimentation. In linear SVM, the kernel used was linear with degree = 1; the C (box constraint) value was one by default. The quadratic SVM was similar to linear SVM, but it used a polynomial kernel with degree = 2. Both the linear and quadratic SVM methods used the one-vs.-one approach for multi-class classification. In bagging, the base

learner was a DT, the number of learners was 30, and the learning rate was 0.1. "*Choosing the size of an ensemble involves balancing speed and accuracy. Larger ensembles take longer to train and to generate predictions. Some ensemble algorithms can become over trained (inaccurate) when too large*" [56]. In [57], the authors suggest using an incremental approach to determine the number of learners, with the goal of choosing the number of learners in which the re-substitution error is minimized. After experimentation, the number of learners was thus set to 30. Boosting had the same parameters as bagging, apart from the ensemble method: boosting used 'RUSBoost', whereas bagging utilized 'Bag'. As discussed in Section 2.4, RUSBoost can handle the class imbalance problem. We next discuss the evaluation techniques commonly used in classification problems.

6.2 Evaluation Measures

The success of a classifier is measured by considering how well it performs in comparison to other state-of-the-art classifiers. A common evaluation criterion in classification problems is the accuracy of the learner, which is discussed in the following section.

6.2.1 Accuracy

Accuracy is an important measure when both positive and negative classes are equally important and an evaluation of system performance as a whole is needed.

The formula for accuracy in [107] is given as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{18}$$

where, TP is true positive, TN is true negative, FP is false positive, and FN is false negative. To avoid over-fitting, a 10-fold cross-validation is used for calculating accuracy when a dataset is small [32]. The entire dataset is split into 10 parts: 9 are used for training while 1 is used for testing. In this way, the test component of an earlier phase becomes part of training in the current phase, and one of the datasets from the training stage of a previous phase becomes a test set in the current period. The average number of accuracies in each fold is calculated to determine final accuracy. Although the accuracy of a system as a whole is needed, it has been argued that accuracy is not the correct measure when classes are

skewed [45, 86]. In [86], the authors suggested that the receiver operating characteristic (ROC) and AUC are better measures for evaluating imbalanced datasets. In contrast, the authors in [45] recommend using the true positive rate (TPR) or false negative rate (FNR) to evaluate skewed datasets; these concepts are discussed below.

6.2.2 True Positive Rate and False Negative Rate

The TPR, which is also known as recall or sensitivity, is the probability that a (randomly selected) relevant image is retrieved in a search [107]; the FNR is the probability that a (randomly selected) relevant image is not retrieved in a search. Equations (19) and (20) are derived from [107].

$$True\ Positive\ Rate\ (TPR) = \frac{(TP)}{(TP+FN)} \tag{19}$$

$$False\ Negative\ Rate\ (FNR) = \frac{(FN)}{(TP+FN)} = 1 - TPR \tag{20}$$

Knowing TPR makes it is easy to calculate FNR, and vice versa.

6.2.3 ROC and AUC

The ROC curve shows the performance of a binary classifier [32]. It is plotted using TPR against the FPR. A perfect classifier will score in top-left corner (TPR=1.0, FPR=0.0), while the worst classifier will score in the bottom-right corner (TPR=0.0, FPR=1.0). A random classifier will have a TPR that is similar to the FPR, i.e. 0.5 for both. Anything above the random classifier is usually considered safe. Another evaluation measure is the AUC [32]. The closer the value of AUC to 1.0, the better the classifier.

6.2.4 Testing for Statistical Significance

A statistical test provides a mechanism for making quantitative decisions about a process or processes. The goal is to determine whether enough evidence exists to “reject” a conjecture or hypothesis about a process. The conjecture is called the null hypothesis. Non-rejection may be a good result if we want to continue to act as if we “believe” that the null hypothesis is true. In our thesis, we formulate the following hypotheses:

- H_0: The performance of the six algorithms is exactly the same.
- H_1: At least one pair of algorithms exists in which performance differs.

In our work, we have six algorithms to evaluate on three domains. In the case of multiple algorithms and multiple domains, two alternatives are generally advisable [36]: the parametric alternative is one-way repeated-measure ANOVA and the non-parametric alternative is Friedman's test.

Because ANOVA assumes a normal distribution, we will use Friedman's test for our statistical test. It works without any data distribution assumptions, by ranking the results for different classifiers separately and computing the sum of the ranks [36]. Friedman's statistics (x_F^2) are calculated as [36]:

$$x_F^2 = \left[\frac{12}{n*k*(k+1)} * \sum_{j=1}^{k}\left(R_j\right)^2\right] - 3 * n * (k+1) \quad (21)$$

where n = number of domains, k = number of algorithms, R_j = rank of an algorithm on domain j, and k-1 is the degree of freedom.

These values are used to look up the p-value in the x^2 distribution table. The p-value is measured against a predetermined threshold value known as the significance level α, which is considered 0.01 or 0.05. It has been standard practice to use 0.05, just to avoid being too stringent while still having good statistical significance.

The null hypothesis, H_0, is rejected if the calculated (x_F^2) value is greater than the p-value obtained from the x^2 table with $\alpha = 0.05$. Rejection of the null hypothesis means that at least one pair of algorithms with unequal performance exists; a post hoc Nemenyi test [66] is conducted to identify the pair(s). The performance Q_{YZ} of two classifiers Y and Z must be greater than the critical difference Q_A to conclude that the performances of classifiers Y and Z are not equal.

6.3 Experimental Results

We begin this section by presenting the results of using a common search dictionary and individual search dictionaries for different categories of pictures. Thereafter we show the

accuracy, TPR, ROC, and AUC values of various learning algorithms on SURF and LAB features to evaluate which algorithm performs better under which conditions. The average of ten runs of 10-fold cross validation were used to capture the results in accuracy, TPR, ROC, and AUC graphs.

Our approach consists of using low-level representations of images and learning multi-class classifiers on the obtained features, with an aim of achieving higher retrieval accuracy for building an efficient CBIR. As discussed in Section 3.2, six semantic categories were identified (Graphics, Machines, People, Landscape, Protest, and Open-Pit); images were thus divided into these six categories.

We implemented three different approaches. The first used the LAB color histogram as a feature space, with a single search dictionary for image indexing and retrieval. The second method also used the LAB feature space for encoding images; however, instead of utilizing a single search vocabulary, six dictionaries (one for each category) were used. The third technique also employed six dictionaries, but in contrast to the second technique (which used LAB color space), it utilized SURF features.

Figure 53. Query image from the Protest category

Figure 53 above contains a query image from the Protest group. It has several features in common with other classes; for instance, it includes people, buildings, and posters. This makes it hard to classify the image without knowing that it actually belongs to the Protest class.

We next present the results obtained from three approaches.

6.3.1 CBIR Using the LAB Color Histogram with a Single Search Dictionary

Figure 54 below contains the results of the first technique, namely CBIR using the LAB color histogram as the feature space with a common dictionary for all pictures and no trained classifiers. When given a query image from Protest group, top 20 images were retrieved based on similarity. We can observe that only 6/20 images were from the actual Protest category, whereas 14 images were from the other categories; this yields a precision rate of 0.3. This low result was expected because all of the images were grouped into a single dictionary regardless of their class-type. Several such experiments on other content-types were conducted and the precision rate obtained by other queries was not more than 0.3.

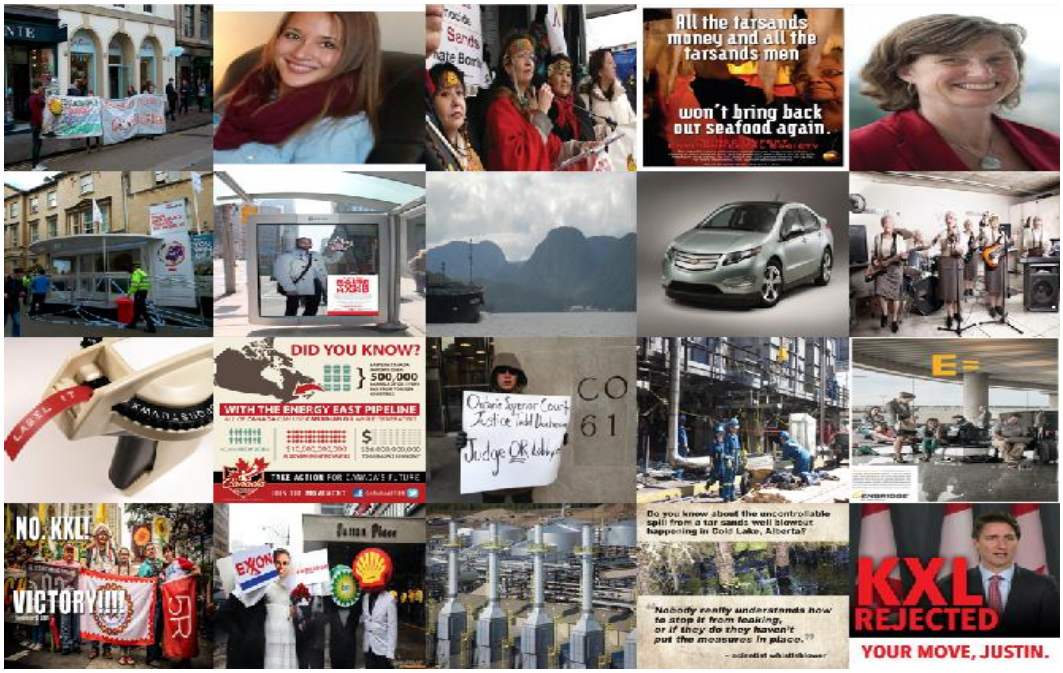


Figure 54. Top 20 results for Protest class using LAB color space with a common image dictionary for all categories

Our next goal was to determine whether creating a separate search dictionary for each category and using trained classifiers improves our results.

6.3.2 CBIR Using the LAB Color Histogram with Each Class Having a Separate Search Dictionary

The second approach we implemented is based on using the LAB color histogram with each content-type having its individual search dictionary. A 10-fold cross validation approach is used to train classification models. Features of query image are given to a 3-step classifier, which predicts the target class of the query image. Based on the target class, a small sub-dictionary is searched for similar images. Figure 55 below shows the outcome of one query where a precision rate of 1.0 was achieved, as 20/20 images were from the Protest category. Several such experiments were conducted, and the results suggest that clustering images in different dictionaries based on their content followed by retrieving pictures from a particular dictionary appears to be a good technique the image collection to be searched is already known (in our case, from one of the different machine learning classifiers). This technique is hence better than LAB color space with a common search dictionary for all categories.

Figure 55. Top 20 results for Protest class using LAB color space with each category having its own separate dictionary

6.3.3 CBIR Using SURF Features with Each Class Having a Separate Search Dictionary

Thirdly, we implemented a CBIR that uses the SURF features approach, with each category having its own separate search dictionary. In Figure 56 below, SURF features are used for the low-level representation of images. Based on the class predicted from the quadratic SVM (best-learned classifier), the Protest image collection was searched. As all of the resultant images were from the Protest category, precision = 1.0.

Figure 56. Top 20 results for Protest class using SURF feature space with each category having its own separate dictionary

Results show that each content-type having its own individual search dictionary can significantly improve retrieval results. However, the collection to be searched for retrieving documents must still be determined.

As we were considering the LAB color space and the SURF features techniques, we further evaluated them to determine which attributes lead to a more accurate classifier. For this exercise, multiple classifiers were separately trained on LAB and SURF features. As discussed in Section 5.2.2, three classification models were prepared. The results of these models are discussed next.

6.3.4 Evaluation: Accuracy of Classifiers on People vs. Non-People

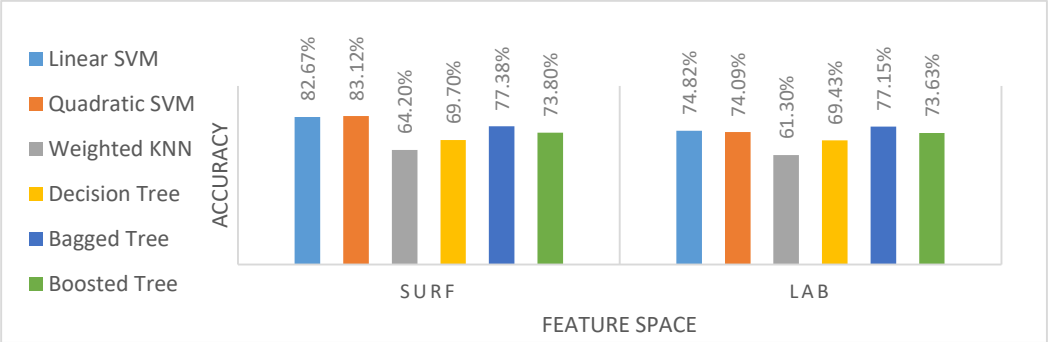


Figure 57. Accuracy of classifiers on SURF and LAB feature space for the People vs. Non-People categories

For this initial experiment, the results in Figure 57 above suggest that SURF features are more accurately classified than LAB features on all of the classifiers. Although quadratic SVM trained on SURF features outperforms all other classifiers, it performs close to linear SVM under both feature spaces. As noted in Section 2.4.1, linear SVM uses a polynomial kernel with a degree of 1, whereas quadratic SVM uses a polynomial kernel with a degree of 2. Although the improvement of quadratic SVM over linear SVM is minimal in relation to our dataset, the results can still be verified in Figure 57 above. Furthermore, bagging has provided better results than boosting in both feature spaces. An accuracy of 83.12% was maximally achieved with SURF features by using a quadratic SVM algorithm, whereas LAB features could only achieve an accuracy of 77.15% (which was obtained by bagged tree classifier). As Bagging and Boosting used DT as a base classifier, their results were poor than SVM.

6.3.5 Evaluation: Accuracy of Classifiers on People vs. Protest

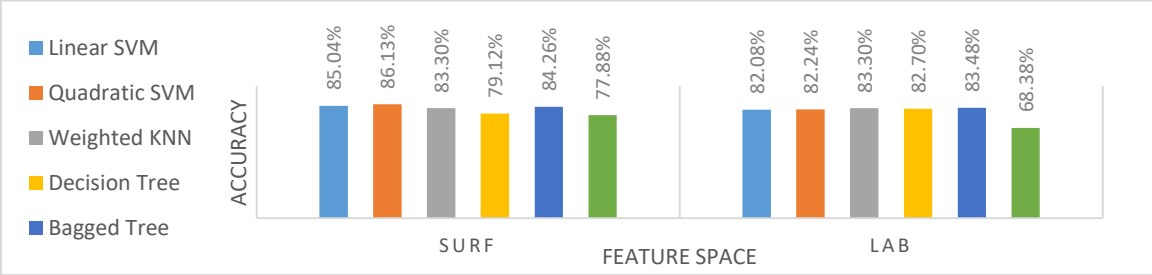


Figure 58. Accuracy of classifiers on SURF and LAB feature space for the People vs. Protest categories

In Figure 58 above, quadratic SVM using a SURF features approach again achieves the best results in terms of overall accuracy. The differences in the performance of all the classifiers were marginal. Under the current domain, People vs. Protest, quadratic SVM achieves better accuracy for SURF features, whereas weighted K-NN is more suited to LAB features. Bagging and boosting, which are known to provide an improvement over a base classifier (DT), have shown unexpected results in overall accuracy. Bagging demonstrated a slight improvement over DT in both feature spaces, whereas the accuracy of the boosting classifier dropped by 2% using the SURF features technique and more significantly by 14% in the LAB color space approach.

6.3.6 Evaluation: Accuracy of the Classifiers for the Machines, Open-Pit, Landscape, and Graphics categories

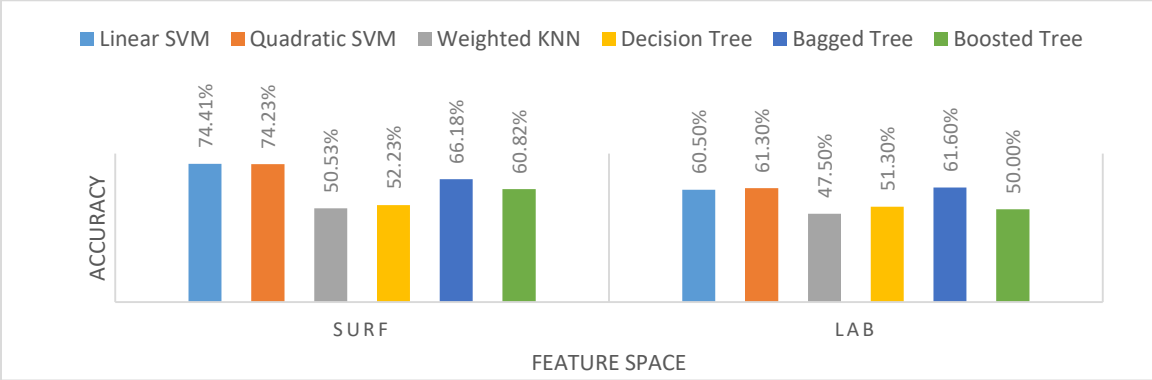


Figure 59. Accuracy of the classifiers on SURF and LAB feature space for the Machines, Open-Pit, Landscape, and Graphics categories

The results in Figure 57, Figure 58, and Figure 59 above suggest that quadratic SVM using a SURF features approach achieves higher accuracy in comparison to all other classifiers, in both binary and multi-class classification. Linear and quadratic SVM have performed similarly in both feature spaces. Furthermore, bagging and boosting improve performance over a simple DT, but their accuracy is still less than SVM.

The next section discusses the TPR of individual categories, to determine how the classifiers perform on individual content-types.

6.3.7 Evaluation: The TPR of People vs. Non-People

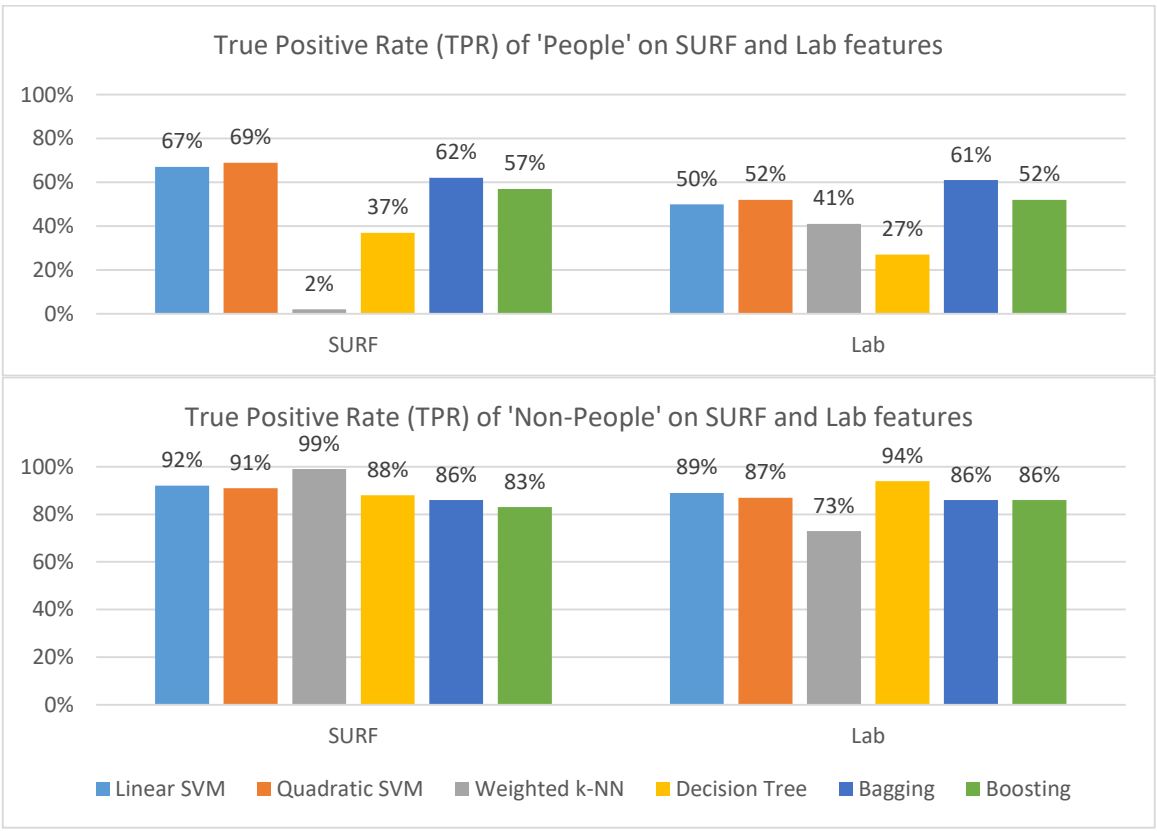


Figure 60. The TPR of all classifiers for People vs. Non-People

In Figure 60, the quadratic SVM classifier has achieved the highest TPR for People (69%) by using a SURF features approach, whereas in LAB features method bagging obtained the maximum TPR (61%) for the People class. In the Non-People category, weighted K-NN achieved the highest TPR (99%) when SURF features were used, and DT achieved the highest TPR (94%) with the LAB features method. Furthermore, it was observed that bagging and boosting improve the performance of the base classifier (DT) when classifying the People class; in contrast, the TPR of bagging and boosting decreased in the Non-People category. The performance of weighted K-NN and DT was unpredictable; while these two algorithms achieved a high TPR in the Non-People category, they achieved a low TPR in the People category. All of the other algorithms performed well in both the categories. The difference in the performance of all of the classifiers on the People and Protest categories may be due to the class imbalance problem, which is addressed in Section 6.3.14.

6.3.8 Evaluation: The TPR of People vs. Protest

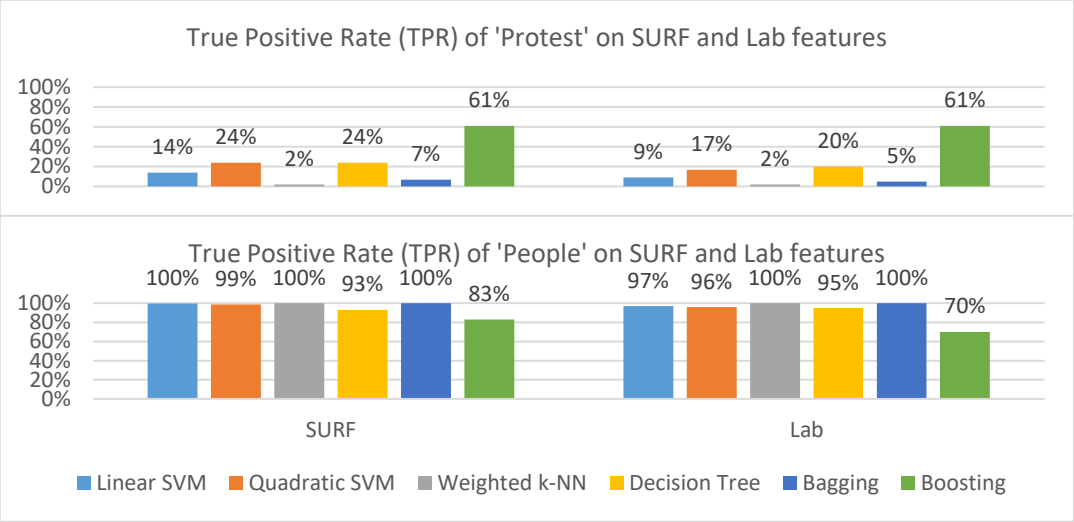


Figure 61. The TPR of all classifiers for People vs. Protest

In Figure 61 above, it was perceived that while all of the classifiers performed equally well in the People class, these algorithms performed equally poorly in the Protest class; the exception is the RUSBoost algorithm, which achieved a TPR > 60% for both classes. The unpredictable behaviour of the classifiers on the two different classes was a consequence of the so-called class imbalance problem, which is covered in Section 6.3.14 below. The performance of all of the algorithms except boosting could not achieve TPR > 24% in the Protest category, whereas the same algorithms achieved more than 93% TPR for the People category using both the SURF and LAB features approaches. Notably, the RUSBoost algorithm achieved the lowest TPR among all classifiers for the People category, whereas the same algorithm achieved the highest TPR among all classifiers for the Protest category.

6.3.9 Evaluation: The TPR of the Machines, Open-Pit, Landscape, and Graphics categories

In Figure 62 below, using the SURF feature approach enabled each classifier to achieve a greater TPR than in the LAB features method. In three out of four classes (Machines, Open-Pit, and Graphics), SVM achieved a high TPR. The performance of weighted K-NN was unpredictable. Using the S-URF features approach for the Landscape content-type, weighted K-NN achieved a TPR of 79%, whereas TPR was only 3% for the LAB features

approach. The RUSBoost algorithm consistently performed better for all datasets. The performance of DT was worse, as it classified almost everything as Graphics. Similarly, weighted K-NN classified every image as either Graphics or Landscape. The other four classifiers were still able to predict the Landscape, Open-Pit, and Machines categories to a certain degree. The Graphics class was responsible for the low TPR in this domain: Graphics overlapped with almost every other category, which made image classification difficult. In the current domain, an SVM classifier trained on SURF features performed better in terms of the TPR. Similar to the previous domains, the current domain also seems to be affected by the data skewness problem. This issue is addressed in Section 6.3.14 below.

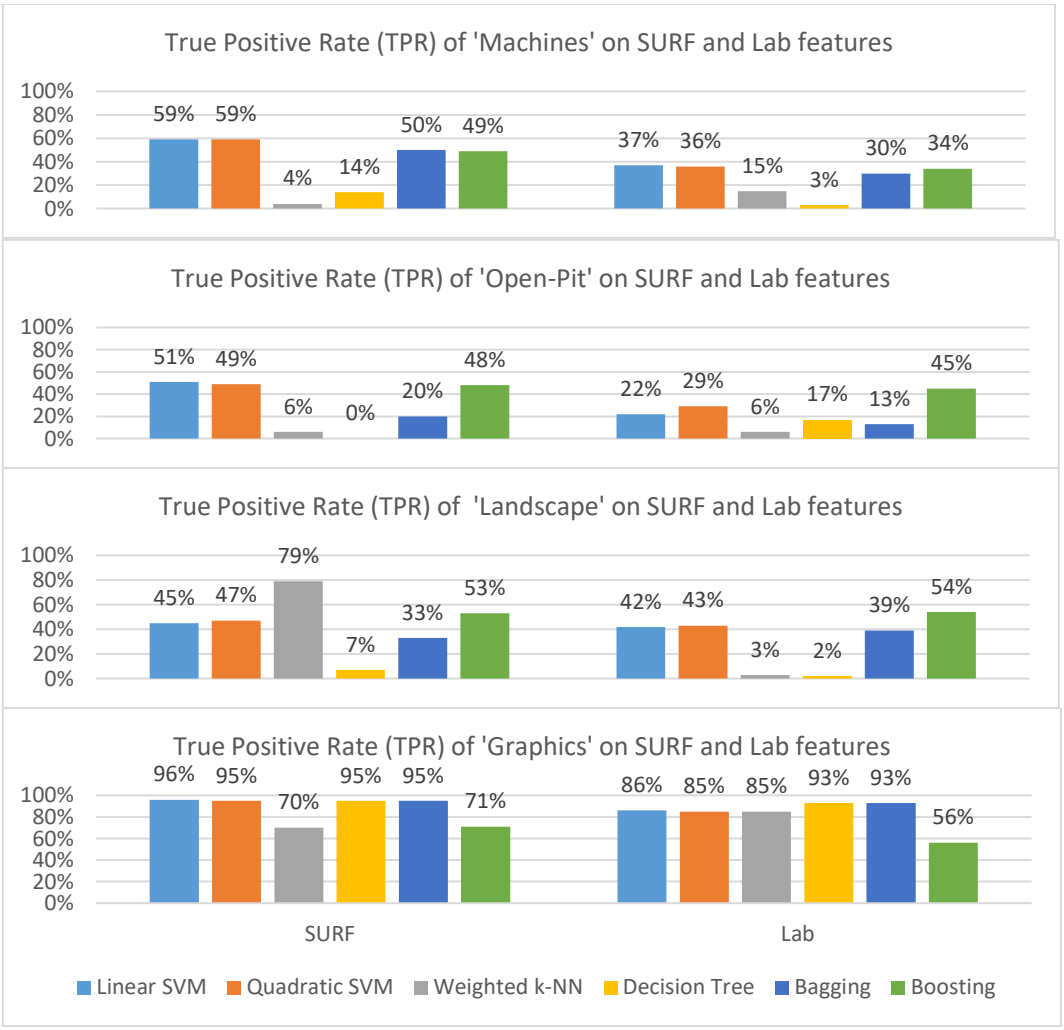


Figure 62. The TPR of all of the classifiers for the Machines, Open-Pit, Landscape, and Open-Pit categories

6.3.10 Evaluation: The ROC and AUC of People vs. Non-People

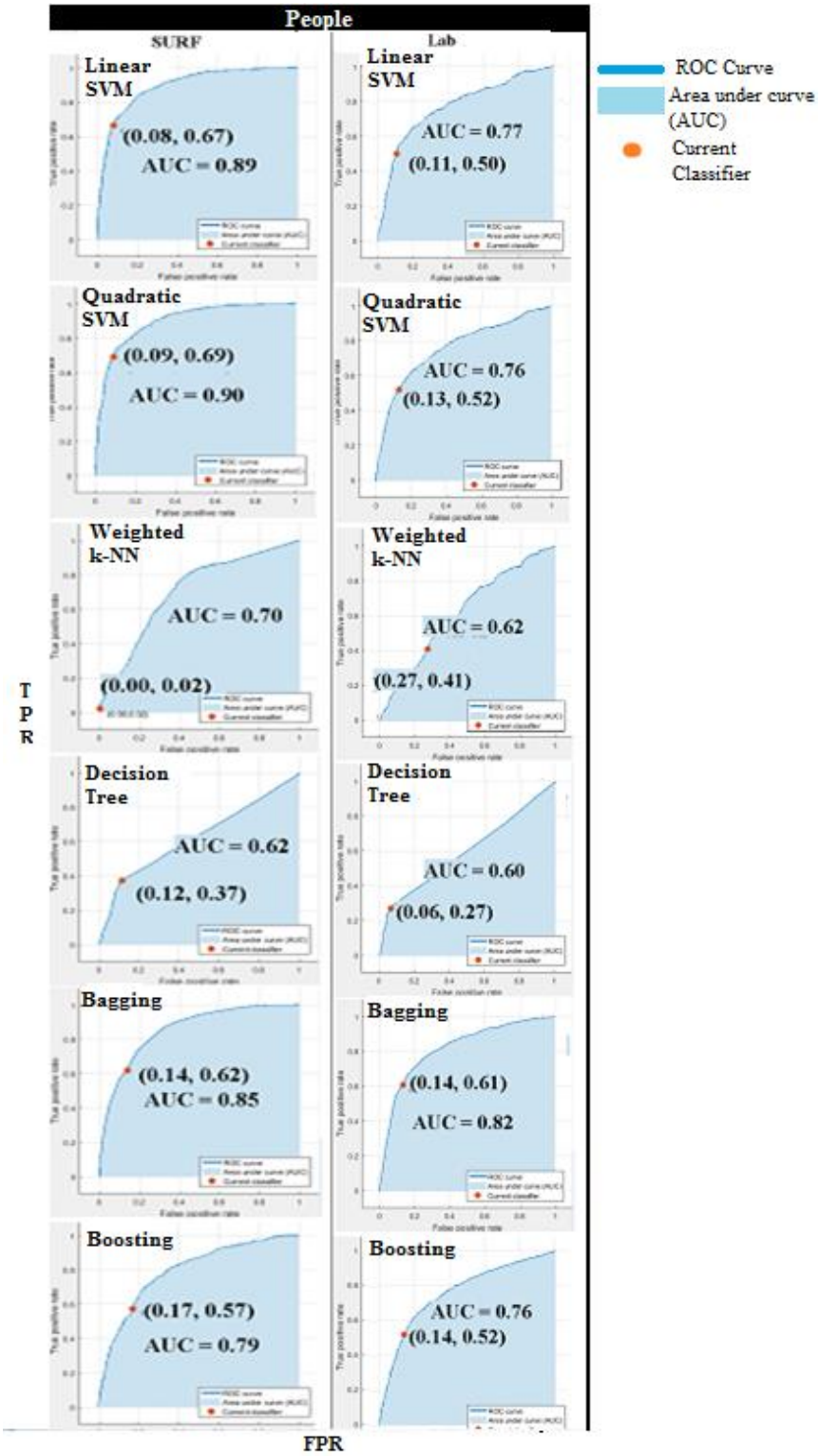


Figure 63. The ROC curve and AUC value for all of classifiers for the People vs. Non-People categories

Figure 63 above shows that all of the classifiers have attained higher AUC values using the SURF features approach in comparison to the LAB features technique. Linear and quadratic SVM have performed similarly for both feature spaces and attained the highest AUC value in the current domain. Another classifier that secured greater AUC is bagging. Although the AUC value of bagging is close to SVM under SURF features, the TPR and FPR values of quadratic SVM (TPR = 0.69, FPR = 0.09) signify that SVM is better than bagging (TPR = 0.62, FPR = 0.14) in the current domain. Quadratic SVM attained the highest AUC (0.90) under SURF features, whereas bagging attained the maximum AUC value (0.82) in LAB features. Under the People content-type, DT performed as the worst classifier for both feature types, where DT reached the lowest AUC values for the SURF features space method (62%) and the LAB color space approach (60%).

6.3.11 Evaluation: The ROC and AUC of People vs. Protest

Figure 64 below illustrates that in four out of six classifiers (linear SVM, quadratic SVM, bagging, and boosting), the SURF features space technique leads to higher AUC values than the LAB color space approach. Similar to the People vs. Non-People domain in Section 6.3.10, linear and quadratic SVM have performed similarly in both feature spaces and have once again achieved the highest AUC value of 0.90 each. Following the two types of SVM, the classifier that achieved the next highest AUC is bagging (0.84). While DT was the worst classifier in Section 6.3.10, here weighted K-NN performs worst with an AUC of 0.58 under both feature spaces and both content-types (People and Protest). Despite having a higher AUC value for the Protest class, quadratic SVM is less preferable than boosting; this is because RUSBoost retrieves a higher TPR rate (0.61) than quadratic SVM (0.24). The higher TPR value achieved by the RUSBoost classifier makes it the best choice for binary classification in this domain. For the People content-type, only boosting achieved a reasonable FPR (0.39); all other classifiers had a minimum FPR of 0.76.

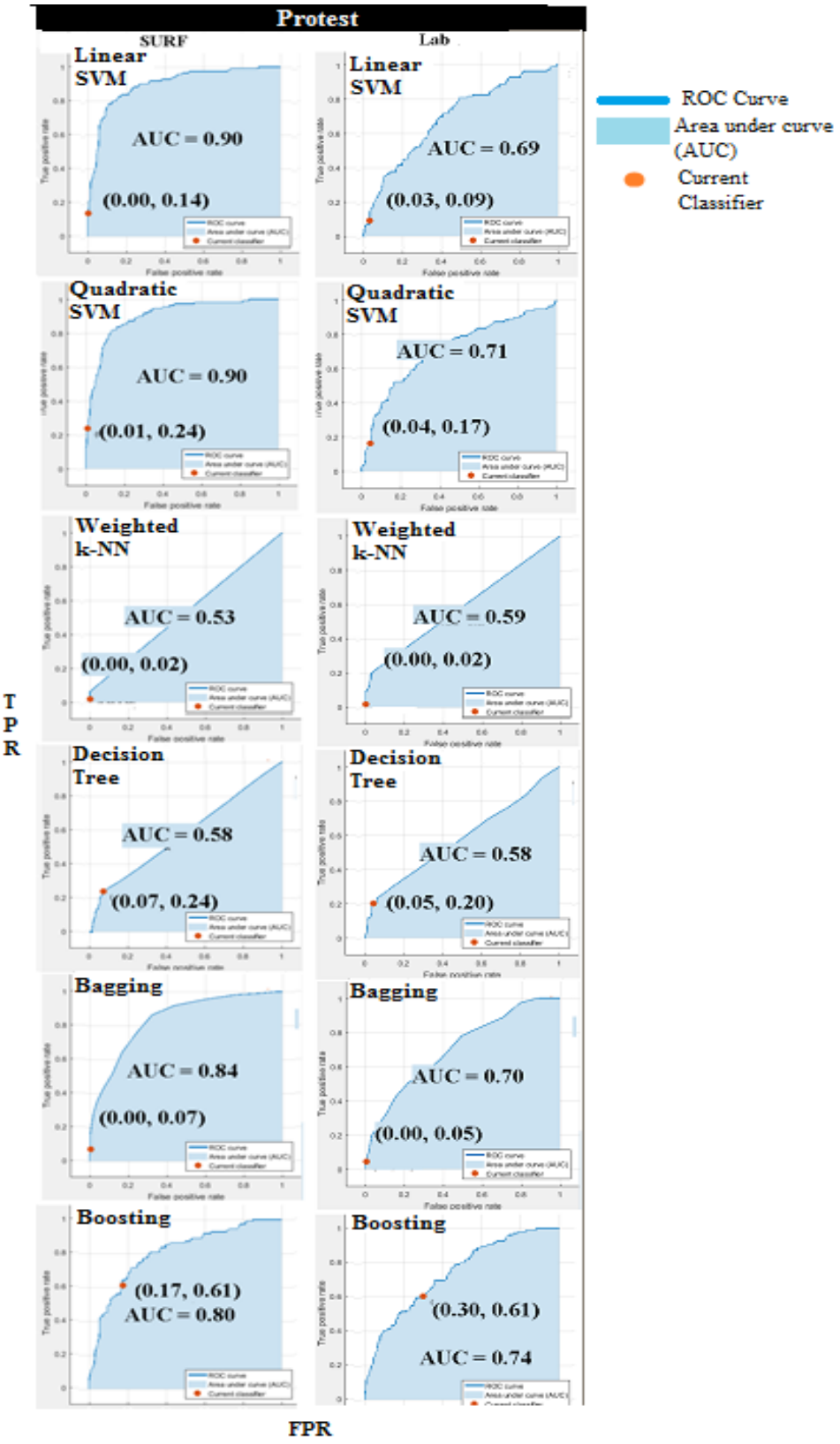


Figure 64. The ROC curve and AUC value for all of classifiers for the People vs. Protest categories

6.3.12 Evaluation: The ROC and AUC of the Machines, Open-Pit, Landscape or Graphics categories

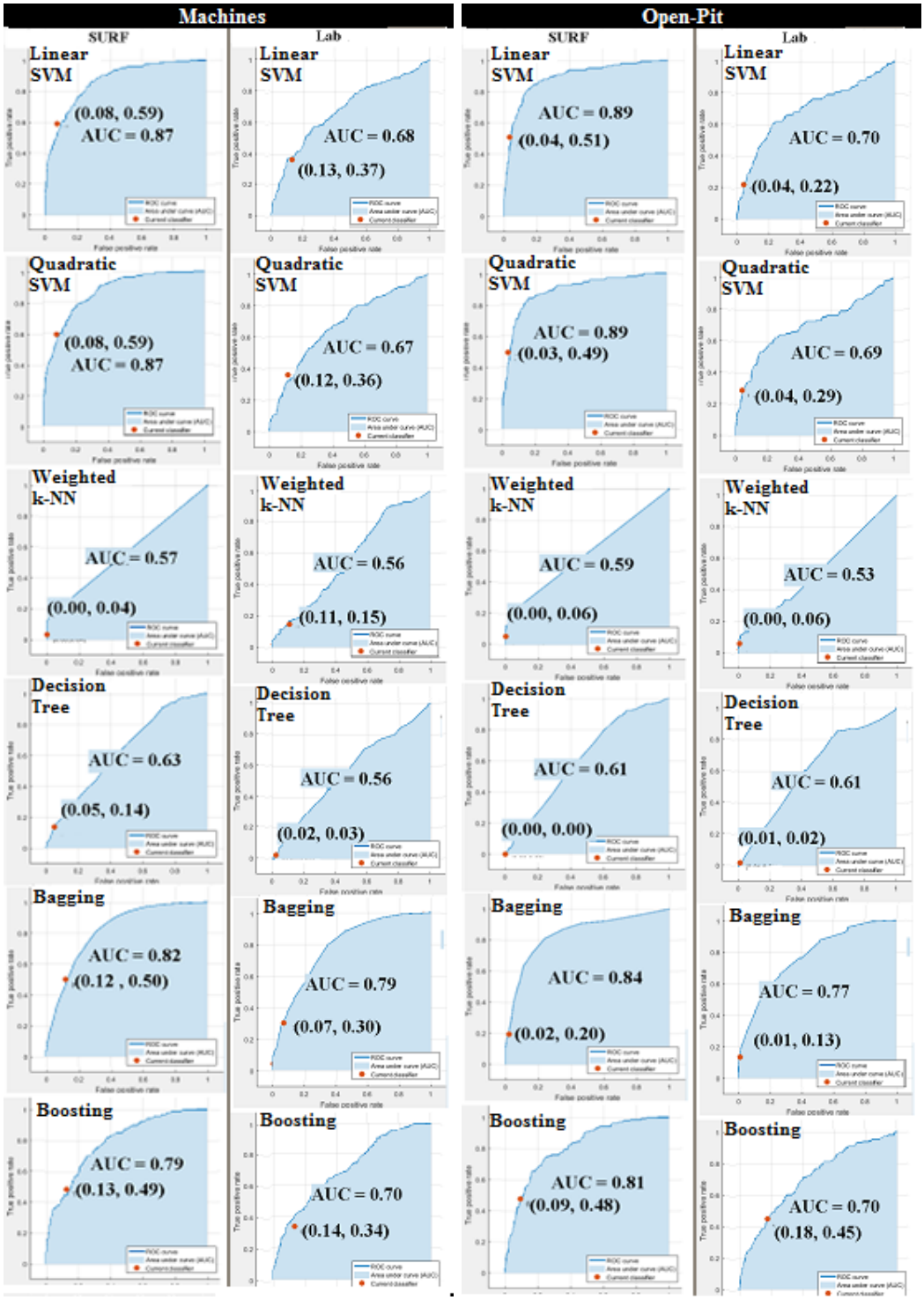


Figure 65. The ROC curve and AUC value for all of the classifiers for the Machines and Open-Pit categories

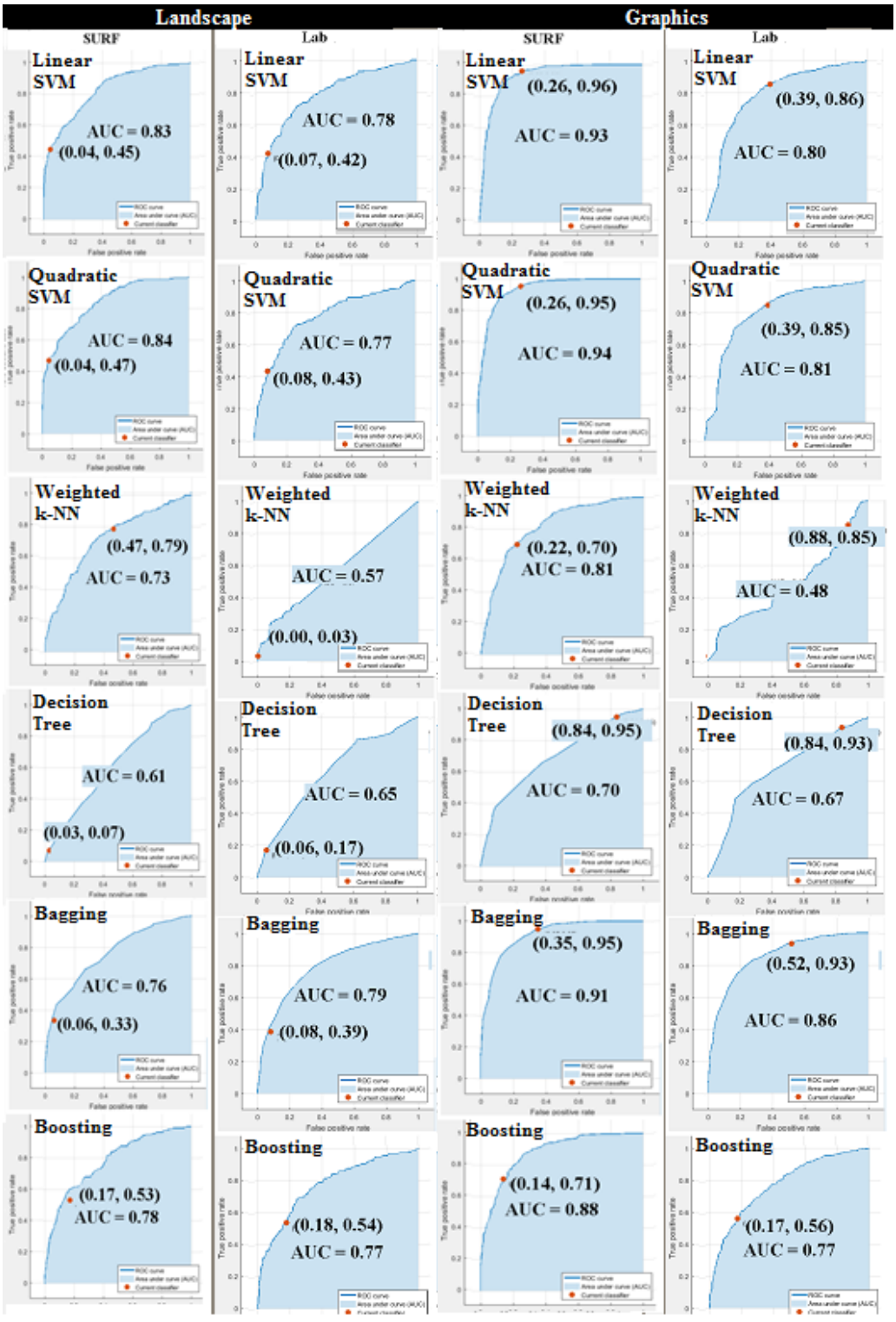


Figure 66. The ROC curve and AUC value for all of the classifiers for the Landscape and Graphics categories

In Figure 65 and Figure 66 above, the SURF features generate higher AUC values than the LAB features for all four types of images; the exception is the Landscape class, where the

LAB features approach leads to higher AUC values for weighted K-NN and DT algorithms. Linear and quadratic SVM used with the SURF features approach achieve the highest AUC (0.89) for the Open-Pit content-types. These two algorithms also performed equally on another class, namely Machines, by achieving an AUC of 0.87. However, quadratic SVM showed a slight improvement in AUC over linear SVM in the Landscape and Graphics classes. The weighted K-NN classifier performed poorly in the Machines and Open-Pit classes, whereas DT achieved the worst AUC in the Landscape and Graphics categories. Notably, boosting generally performed better than bagging in the current domain by predicting more true positives for all classes except Graphics. In contrast, SVM was showed equal or better TPR values or all four classes using the SURF features approach.

6.3.13 The Statistical Significance of the Accuracy Results

It has been noted that we used Friedman's test to check the statistical significance of our accuracy results. As explained in Section 6.2.4, a threshold value of $\alpha = 0.05$ was used as our significance level. Two experiments were conducted: the first used the SURF features method, the second used the LAB color space approach. Here domain 1 included images from the People and Non-People classes, domain 2 comprised People and Protest images, and domain 3 contained pictures from the Machines, Landscape, Open-Pit, and Graphics categories. The table below presents the rankings from the Friedman's test for the SURF features approach across the three domains.

Table 10. Friedman ranking for SURF features

	Linear SVM		**Quadratic SVM**		**Weighted KNN**		**DT**		**Bagged Tree**		**Boosted Tree**	
Domain1	82.67%	5	83.12%	6	64.20%	1	69.70%	2	77.38%	4	73.80 %	3
Domain2	85.04%	5	86.13%	6	83.30%	3	79.12%	2	84.26%	4	77.88%	1
Domain3	74.41%	6	74.23%	5	50.53%	1	52.23%	2	66.18%	4	60.82%	3

The following values were obtained using the above table:

$$x_F^2 = 13.09, \qquad k = 6, \qquad n = 3, \qquad p\_value = 9.85$$

The $x_F^2 > p\_value$ signifies that the null hypothesis does not hold, which means that all algorithms are not equal on the basis of accuracy. A post hoc Nemenyi test was then conducted to determine which pair of algorithms was different.

Table 11. Q_{YZ} values for SURF features from the post hoc Nemenyi test

	Linear SVM	Quadratic SVM	Weighted KNN	DT	Bagged Tree
Quadratic SVM	0.65				
Weighted KNN	**7.20**	**7.86**			
DT	**6.55**	**7.20**	0.65		
Bagged Tree	2.62	3.27	**4.58**	**3.93**	
Boosted Tree	**5.89**	**6.55**	1.31	0.65	3.27

The critical value of $Q_A = 3.47$ was obtained from Tukey's table. The p-values for each pair of algorithms A and B were obtained from tables of Q that were especially prepared for the Friedman test [109]. Section 6.2.4 noted that Q_{YZ} rejects the null hypothesis when $Q_{YZ} > Q_A$. The p-value = 7.86 of the pair quadratic SVM vs. weighed K-NN is highly significant, as it is greater than $Q_A = 3.47$. This implies that quadratic SVM and weighed K-NN do not perform similarly and that these two algorithms reject the null hypothesis that they are equal on the basis of accuracy. The strongest classifier is SVM with a quadratic kernel, and the weakest performance was delivered by weighted K-NN. Similarly, other pairs that reject the null hypothesis are linear SVM vs. weighted K-NN, linear SVM vs. DT, linear SVM vs. boosted tree, quadratic SVM vs. DT, quadratic SVM vs. boosted tree, weighted K-NN vs. bagging, and DT vs. bagging. The results signify that the performance of linear SVM is statistically different from weighted K-NN, DT, and boosting, where linear SVM is better than its counterparts. Similarly, quadratic SVM is statistically better than weighted K-NN, DT, and boosting. When the performance of weighted K-NN vs. bagging is considered, bagging is revealed to be statistically superior to weighted K-NN. Similarly, bagging is also superior to DT.

Another set of experiments was conducted on the LAB color space. The table below contains the rankings from the Friedman's test for LAB features across the three domains.

Table 12. Friedman ranking for LAB features

	Linear SVM		Quadratic SVM		Weighted KNN		DT		Bagged Tree		Boosted Tree	
Domain1	74.82%	5	74.09%	4	61.30%	1	69.43%	2	77.15%	6	73.63%	3
Domain2	82.08%	2	82.24%	3	83.30%	5	82.70%	4	83.48%	6	68.38%	1
Domain3	60.50%	4	61.30%	5	47.50%	1	51.30%	3	61.60%	6	50.00%	2

The following values are obtained from the above table:

$$x_F^2 = 8.90, \qquad k = 6, \qquad n = 3, \qquad p\_value = 9.857$$

The $x_F^2 < p\_value$ signifies that the null hypothesis holds, which indicates that all algorithms are equal on the basis of accuracy. Conducting a post hoc Nemenyi test is therefore unnecessary.

6.3.14 Evaluation of the Minority Classes

In Table 13 below, it was noticed that all three domains are, to some extent, subject to the so-called class imbalance problem. In domain 1, the People category has fewer instances, and the ratio of the two categories is 1:2. Similarly, in domain 2, Protest is the minority class and the ratio of the two classes is 5:1. Domain 3 contains three minority classes (Machines, Open-Pit, and Landscape) and the ratios of these categories' instances to the Graphics category are 1:2.6, 1:4, and 1:3 respectively. Although quadratic SVM attained the highest accuracy (using both SURF and LAB features), we further evaluated the classifiers to determine their behaviour on the minority classes in all three domains.

Table 13. Number of instances in the three domains

	Domain 1		Domain 2		Domain 3			
	People	Non-People	People	Protest	Machines	Open-Pit	Landscape	Graphics
# Instances	642	1118	533	109	216	142	192	568

In domain 1 (People vs. Non-People), the minority class is People. Upon closely examining the TPR of this minority (as shown in Figure 60 above), we found that quadratic SVM achieves the highest TPR (69%), followed by linear SVM (67%). RUSBoost, which is known to handle the class imbalance problem, successfully classified minority class by achieving a TPR = 57%. It was observed from the dataset that if the classes are skewed to a ratio of 1:2, quadratic SVM achieves more true positives than boosting.

In domain 2 (People vs. Protest), the class that has fewer instances is Protest. The ratio of instances in two categories is 5:1, and the dataset is considered to be highly skewed. In Figure 61, it was observed that only RUSBoost was able to handle the class imbalance

problem; it secured a TPR of 61% for the Protest category. No other classifier was able to secure a TPR more than 24%. The results suggest that RUSBoost achieves the higher TPR for minority classes, which it does by compromising the performance of the majority class.

Domain 3 (Machines vs. Open-Pit vs. Landscape vs. Graphics) has three minority classes, namely Machines, Open-Pit, and Landscape. The results in Figure 62 suggest that linear SVM, quadratic SVM, and RUSBoost performed similarly for all three minority classes and were successful in identifying the minority classes. The results also propose that linear and quadratic SVM do not degrade the performance of the majority class, whereas RUSBoost slightly decreases it in moderately skewed datasets.

6.4 Training Time Analysis

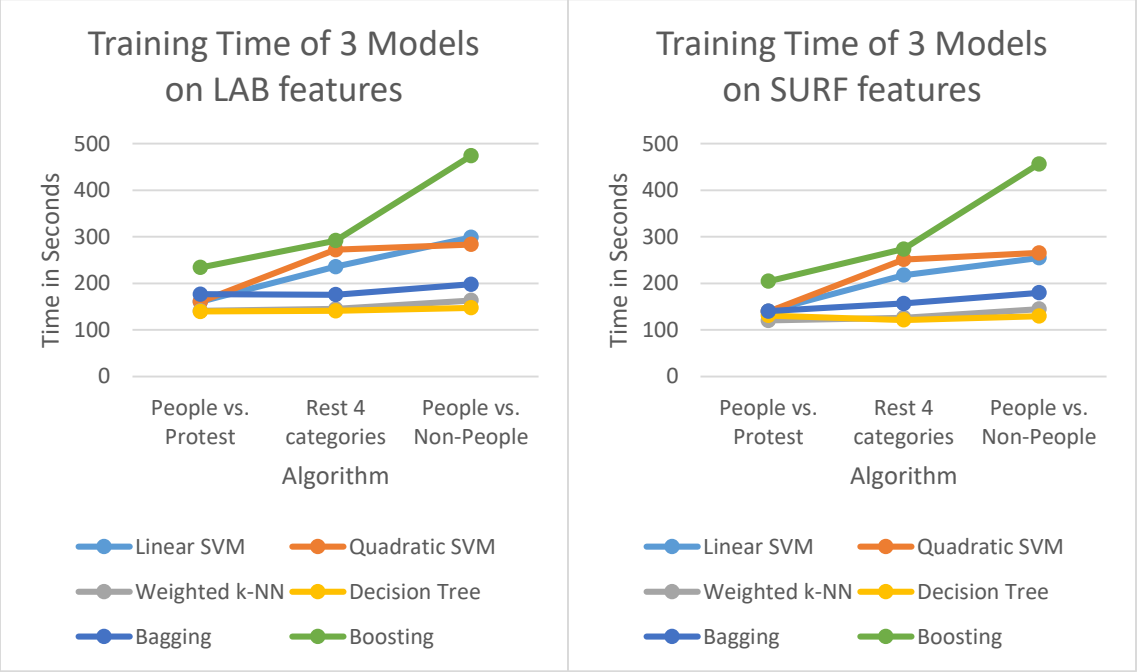


Figure 67. Training Time required by six algorithms for training three models on LAB and SURF features.

Figure 67 above shows the time required by different machine learning algorithms to learn classifiers on three given domains. People vs. Protest domain has least number of instances, i.e., 642 instances. Following this, rest four categories has 1118 instances. People vs. Non-People has maximum number of instances, i.e., 1760 training examples. Under both feature-sets, SURF and LAB, Boosting algorithm has taken the highest time for training, followed by SVMs'. It can be observed that training time required by Boosting, Linear

SVM, and Quadratic SVM is proportional to number of instances. As the number of instances increases, the training time also increases. Furthermore, the rate of growth of training time, as the number of instances increases, required by Weighted k-NN, Decision Tree, and Bagging algorithms is less in comparison to Boosting and SVMs'.

6.5 Synthesis and Lessons Learned

The goal of this experimental evaluation was to discover the behaviour of different machine learning algorithms on low-level features for creating an efficient CBIR system that can achieve a high accuracy of retrieval. Supported by evidence from the literature, as stated in Section 2.3.2.6, the SURF and LAB features were found suitable for image classification problem. Results can be verified from Table 14, where three models were learned using different classifiers. All of the classifiers, apart from DT on domain 2, achieved greater accuracy using a SURF feature as opposed to a LAB feature. The performance of DT was quite similar on both feature types, and it is only a shred of evidence that proves that DT does not completely utilize training data (as specified in Section 2.4).

Table 14. The accuracy of different classifiers on LAB and SURF feature

Sr. No	Models	Feature Type	Linear SVM	Quadratic SVM	Weighted KNN	DT	Bagged Tree	Boosted Tree
1	**People Vs. Non-People**	SURF	82.67%	83.12%	64.20%	69.70%	77.38%	73.80%
		Lab	74.82%	74.09%	61.30%	69.43%	77.15%	73.63%
2	**People Vs. Protest**	SURF	85.04%	86.13%	83.30%	79.12%	84.26%	77.88%
		Lab	82.08%	82.24%	83.30%	82.70%	83.48%	68.38%
3	**Machines/Landscape/ Open-pit/Graphics**	SURF	74.41%	74.23%	50.53%	52.23%	66.18%	60.82%
		Lab	60.50%	61.03%	47.50%	51.30%	61.60%	50.00%

The results in Figure 68 below suggest that beyond classification, SURF features can also be used to find point correspondences. They also indicate that LAB features were unable to handle point correspondence, which means they cannot match keypoints within an image. In contrast, SURF features were able to successfully match keypoints and find point correspondences.

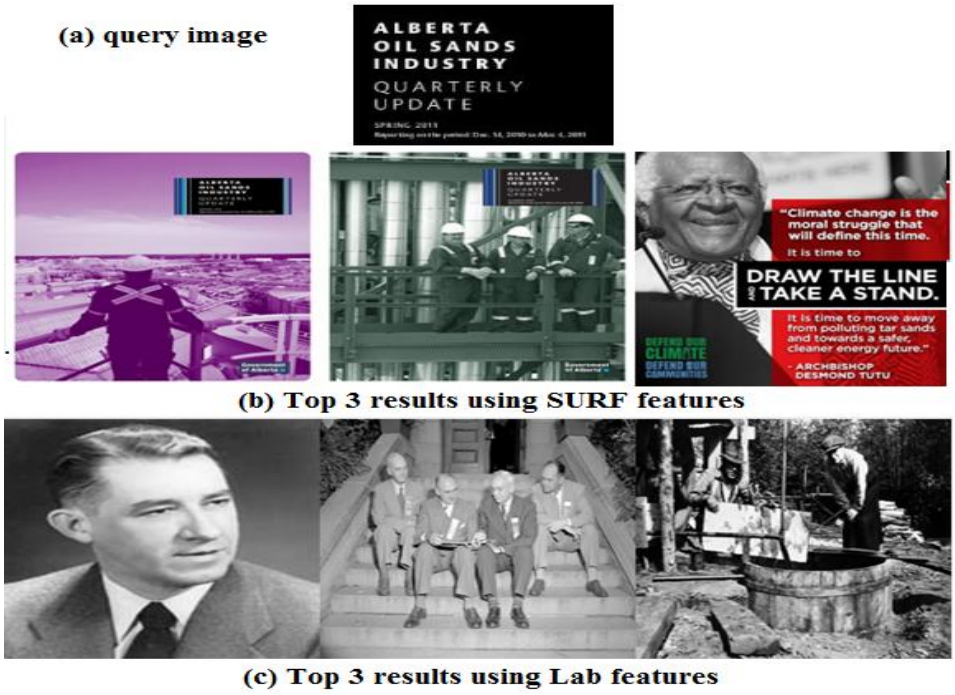


Figure 68. A concept showing point correspondence. (a) Query image, (b) top three results using SURF, and (c) top three results using LAB

In our preliminary experiments (Sections 6.2.1 to 6.2.3), we showed the results we obtained from using both a single search dictionary and individual search dictionaries for each category (People, Protest, Graphics, Machines, Open-Pit, Landscape). We learned that creating separate dictionaries yielded more relevant image retrieval results, as pictures are searched within a particular collection.

With the help of statistical tests, it was perceived in Section 6.2.13 that the SURF features space approach results in all classifiers performing statistically different from one another at a significance level of $\alpha = 0.05$. Furthermore, this approach leads to the quadratic SVM method outperforming all other classifiers in terms of accuracy. We believe that this is due to the approach's properties of efficiently handling high dimensional data and having maximum-margin hyperplanes, which help to separate data to a greater extent [111].

As expected, we discovered that the boosting method we used (RUSBoost) was the best method for higher skewed datasets. On the other hand, the SVM and bagging methods perform poorly against highly imbalanced datasets, while having some success against

moderately skewed data. However, the three methods handle the minority class differently. It was observed that if the dataset is highly imbalanced, only RUSBoost can obtain reliable performance vis-à-vis the minority class. Nonetheless, if the dataset is moderately imbalanced, even SVM and bagging can provide good classification results. In the moderately imbalanced dataset, SVM not only performs well on the minority class; it also maintains the performance of the majority class. In contrast, bagging and boosting improve the performance of the minority class at the cost of decreased majority class performance. Furthermore, boosting was found more reliable and consistent than bagging on any imbalanced dataset. This observation needs to be explored in our future work; we also plan to consider sampling methods [5, 6, 11, 15, 33, 42, 102].

The weighted K-NN technique performed worst out of all of the classifiers. As stated in Section 2.4, the class label assigned to a query image is determined by the majority vote amongst its k-nearest neighbours. In two-class problems such as in domains 1 and 2, this leads to a problem when classes are moderately or highly imbalanced. The majority voting process also affects multi-class problems such as in domain 3, regardless if skewness is high or moderate. The classification performance of weighted K-NN in multi-class problems is further affected by how distinct the categories are from one another. If the instances from different classes overlap, the prediction result of weighted K-NN will deteriorate. In domain 3, pictures from the Machines and Open-Pit categories shared many features with the Graphics and Landscape classes; as a result, weighted K-NN classified almost everything as Graphics or Landscape. Table 15 below again summarizes how the categories were skewed in our database.

Table 15. Number of instances in each category

People		**Non-People**			
People	**Protest**	**Machines**	**Landscape**	**Open-Pit**	**Graphics**
533	109	216	192	142	568

The images in each category overlapped with other classes; the confusion matrix in Figure 69 below shows the result, which is obtained by running a Quadratic SVM algorithm to train a multi-class classifier and computing its TPR and FNR. The People and Protest categories were the most similar, as they both contained humans; 66% of the Protest images

were misclassified as People due to the categories' similar characteristics. Graphics and Landscape were at the other end of the scale, given that Graphics contained computer-assisted visuals (including text), whereas Landscape mostly included scenic nature photographs. As the characteristics of these two classes were quite distinct, not even a single instance of Graphics was misclassified as belonging to Landscape. Graphics and Landscape were joined by Protest and Open-Pit in the least similar category.

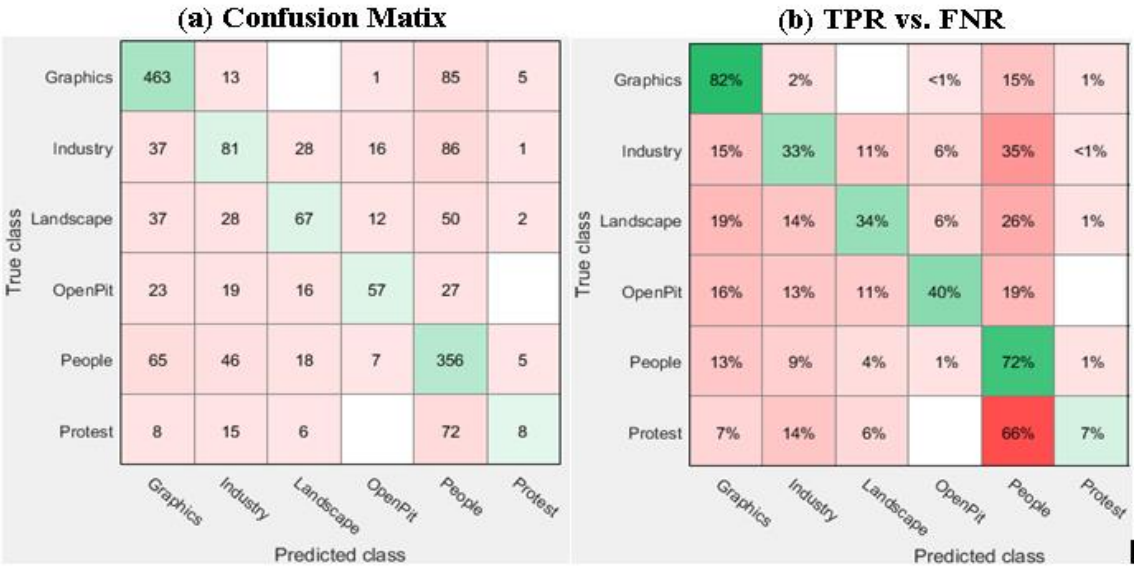


Figure 69. The (a) Confusion Matrix, and (b) TPR and FNR of quadratic SVM on six content-types

Another important observation was that where categories are not distinct, dividing the problem of learning a multi-class classifier into sub-problems could improve the accuracy results. The results can be verified from Table 16 below, which shows an increase in TPR for each content category. The exception is the People category, where the TPR decrease is only 1% and less significant in comparison to improvements in the other domains.

Table 16. The increase in TPR when using a three-step approach for quadratic SVM

Content-Type	**Graphics**	**Machines**	**Landscape**	**Open-Pit**	**People**	**Protest**
Single-Step Approach (TPR)	82%	33%	34%	40%	72%	7%
Three-Step Approach (TPR)	95%	59%	47%	49%	69%	24%

We also discovered that the Protest category was the hardest to learn because it shared properties with the People class. It was also affected by the class imbalance problem, as

People had 533 instances and Protest had only 109. The Graphics class was the easiest to learn, due to the abundant number of instances for training a good classifier.

6.6 Conclusion

In this chapter, we discussed the experimental settings, various evaluation measures, and the results obtained in our experimentation. We started our experiments with the aim of evaluating the use of different classifiers on low-level image features for building a CBIR with greater retrieval accuracy. We discovered that SURF features learned by quadratic SVM is the best choice for creating Mediatoil-IR. The results were validated using a statistical test. Quadratic SVM also performed well on LAB features, but the results were not statistically different from other classifiers. We used the knowledge attained in this chapter for the final implementation of the Mediatoil-IR system and thus were able to realize the concept of creating Mediatoil-IR for exploring the oil sand images.

The next chapter presents a general conclusion of this thesis, as well as our contributions and future work.

Chapter 7

Conclusions

In this thesis, we explored the Alberta oil sands imagery that is used by different stakeholders in their campaigns and debates based on its content. We also evaluated the performance of different machine learning classifiers on various low-level features to create an efficient CBIR system. The aim of this Mediatoil-IR system was to serve as a medium to retrieve archived tar sand pictures. This chapter provides an overall conclusion to the thesis. We discuss both our contributions and our future work.

7.1 Thesis Contributions

There are numerous applications for CBIR systems in different domains, including healthcare, remote sensing, history, fashion, security, crime prevention, biodiversity information systems, publishing, art collections, retail catalogues, medical information retrieval, face finding, and architectural and engineering design [43]. Up to now, researchers in the oil sands domain have mostly used manual approaches when conducting their studies; no one has made use of image processing or machine learning in relation to oil sand documents.

The dataset, which is novel in its own right, was constructed as a part of this thesis to explore media available in the oil sands debates. Data collection for the Mediatoil database covered the period from 1967 to 2015 and aimed to include all on- and offline documents, videos and images available on different stakeholders' websites or any other accessible media. Various stakeholder organizations were identified, each with its own multiple

campaigns and multiple documents. The complete database is available and accessible to the public (through the Mediatoil website) for future research endeavours.

To differentiate the images based on their content, we classified the pictures into six content-types (People, Machines, Landscape, Open-Pit, Protest, and Graphics). These categories were identified after rigorous experimentation and discussions with experts. To train classifiers on images, we needed to identify different low-level representations of images. After comparing various low-level features, we decided to use SURF and LAB features in our experiments. In comparison to the LAB features, the SURF features were found more useful for classification and point correspondence tasks. As a result, we utilized the SURF features for the final implementation of the Mediatoil-IR system.

Deciding on the machine learning algorithm to use for classification was difficult and reminded us of the "no free lunch" [9] theorem. Six different classifiers were ultimately experimented with to identify the best for creating an efficient CBIR. It was concluded that the SURF method trained on quadratic SVM was the best option for creating our image retrieval system, in which achieving higher accuracy is the system's goal. It was also observed that for higher imbalanced datasets, RUSBoost achieves higher minority class TPR values than other techniques. In contrast, it was found that even the SVM and bagging techniques can handle moderately imbalanced datasets.

Based on the content-type identified by machine learning, we also analyzed and differentiated the pictures used by the various stakeholders in their campaigns and debates. A complete chapter was dedicated to this analysis.

7.2 Future Work

At the primary level, our research included images from Reports, Photographs, Still Advertisements, and Factsheets produced by different stakeholders; videos were not analyzed and are left for future research. An independent analysis of document text within reports could also be conducted.

We differentiated the images based on their content, but the classes chosen for image content analysis were found to be overlapping. A new and more distinctive set of categories needs to be identified and evaluated. A different approach to image analysis could also be assessed, in which more than one content-type can be found within the pictures.

We acknowledge from the literature that RUSBoost can handle highly skewed datasets. In addition, SVM and Bagging techniques could also handle moderately skewed datasets. However, as noted in the previous chapter, the performance of various classifiers may be further enhanced by experimenting with using inter alia under- and oversampling techniques to handle imbalanced datasets. Under-sampling removes some instances of the majority class and thus may lead to a loss of information, whereas over-sampling generates artificial samples for the minority class. The various techniques for handling an imbalance are addressed in [5, 6, 11, 15, 33, 42, 102].

It was noted that we utilized K-means clustering for performing quantization while preparing a BOW, using Matlab. In the future, we aim to create our own BOW implementation, in which we can evaluate different clustering techniques.

Another scope for improvement is to introduce active learning [81]. The key idea behind active learning is that "*a machine learning algorithm can achieve greater accuracy with fewer training labels if it is allowed to choose the data from which it learns. An active learner may pose queries, usually in the form of unlabeled data instances to be labeled by an oracle (e.g., a human annotator)*" [81]. In this study, all of the available instances were manually labelled with the help of an expert from the communications department. Both time and effort could be reduced if active learning were involved.

Further improvements could be made by combining the text contained in each image with its low-level features [43, 79, 98, 103]. Various optical character recognition (OCR) algorithms can be used to extract text from the pictures; OCR "*is a process that allows converting scanned or photographed images of typewritten or printed text into editable text*" [24]. The text and low-level features could potentially be combined to achieve greater accuracy of retrieval. For instance, Protest images retrieved from CBIR, can be further filtered in pro or against oil sands by examining their metadata retrieved from OCR.

References

[1] S. S. Ajeesh, M. S. Indu, and E. Sherly. Performance analysis of classification algorithms applied to caltech101 image database. In *Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on*, pages 693–696, Feb 2014.

[2] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346 – 359, 2008. Similarity Matching in Computer Vision and Multimedia.

[3] Herbert Bay, Tinne Tuytelaars, and Luc Gool. *Computer Vision – ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I*, chapter SURF: Speeded Up Robust Features, pages 404–417. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[4] M.J.A. Berry and G.S. Linoff. *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management*. Wiley technology publication. Wiley, 2004.

[5] R. Blagus and L. Lusa. Class prediction for high-dimensional class-imbalanced data. *BMC Bioinformatics*, 11, 2010.

[6] Rok Blagus and Lara Lusa. Smote for high-dimensional class-imbalanced data. *BMC Bioinformatics*, 14(1):1–16, 2013.

[7] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140.

[8] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[9] Eric Cai. Machine learning lesson of the day â€“ the â€œno free lunchâ€• theorem. http://www.statsblogs.com/2014/01/25/machine-learning-lesson-of-the-day-the-no-free-lunch-theorem/, January 2014.

[10] CAPP. Economic contribution. http://www.canadasoilsands.ca/en/explore-topics/economic-contribution, July 2016.

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *J Artif Intell Res*, 16, 2002.

[12] colorbasics.com. Colorspace. http://www.colorbasics.com/ColorSpace/, June 2016.

[13] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297.

[14] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005.

[15] S. Daskalaki, I. Kopanas, and N. Avouris. Evaluation of classifiers for an uneven class distribution problem. *Appl Artif Intell*, 20, 2006.

[16] dct research. The corel database for content based image retriev. https://sites.google.com/site/dctresearch/Home/content-based-image-retrieval, August 2016.

[17] Alberto Del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[18] I. El-Naqa, Yongyi Yang, N. P. Galatsanos, R. M. Nishikawa, and M. N. Wernick. A similarity learning approach to content-based image retrieval: application to digital mammography. *IEEE Transactions on Medical Imaging*, 23(10):1233–1244, Oct 2004.

[19] Student Energy. Oil sands 101. https://www.studentenergy.org/topics/oil-sands-mining?gclid=CKDW7_3Pi8wCFQYNaQodgfcPig#reference-2, May 2015.

[20] Suncor Energy. Osqar, oil sands question and response. http://osqar.suncor.com/, July 2016.

[21] EthicalOil.org. 2012 opec lobbyist of the year. http://www.ethicaloil.org/2012-opec-lobbyists-of-the-year/, August 2016.

[22] EthicalOil.Org. Chiquita conflict. http://www.ethicaloil.org/chiquita-conflict/, July 2016.

[23] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3):231–262.

[24] K. El Gajoui and F. Ataa Allah. Optical character recognition for multilingual documents: Amazigh-french. In *Complex Systems (WCCS), 2014 Second World Conference on*, pages 84–89, Nov 2014.

[25] Alberta Government. Home - let's talk royalties - alberta's royalty review panel. https://letstalkroyalties.ca/, August 2016.

[26] Amarnath Gupta and Ramesh Jain. Visual information retrieval. *Commun. ACM*, 40(5):70–79, May 1997.

[27] J. Gutierrez-CÃ¡ceres, C. Portugal-Zambrano, and C. BeltrÃ¡n-CastaÃ±Ã³n. Computer aided medical diagnosis tool to detect normal/abnormal studies in digital mr

brain images. In *2014 IEEE 27th International Symposium on Computer-Based Medical Systems*, pages 501–502, May 2014.

[28] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, Nov 1973.

[29] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[30] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[31] M. Hassaballah, Aly Amin Abdelmgeid, and Hammam A. Alshazly. *Image Feature Detectors and Descriptors : Foundations and Applications*, chapter Image Features Detection, Description and Matching, pages 11–45. Springer International Publishing, Cham, 2016.

[32] Trevor J. Hastie, Robert John Tibshirani, and Jerome H. Friedman. *The elements of statistical learning : data mining, inference, and prediction*. Springer series in statistics. Springer, New York, 2009. Autres impressions : 2011 (corr.), 2013 (7e corr.).

[33] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Trans Knowledge Data Eng*, 21, 2009.

[34] Marko Heikkilä, Matti Pietikäinen, and Cordelia Schmid. Description of interest regions with local binary patterns. *Pattern Recogn.*, 42(3):425–436, March 2009.

[35] Jing Huang, S. R. Kumar, M. Mitra, Wei-Jing Zhu, and R. Zabih. Image indexing using color correlograms. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 762–768, Jun 1997.

[36] N. Japkowicz and M. Shah. *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, 2011.

[37] Hendrik Drachsler Erik Duval Katrien Verbert, Nikos Manouselis. Dataset-driven research to support learning and knowledge analytics. *Educational Technology & Society*, 15(3):133–148, June 2012.

[38] R. Kimmel, C. Zhang, A. Bronstein, and M. Bronstein. Are mser features really interesting? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2316–2320, Nov 2011.

[39] Santhana Krishnamachari and Mohamed Abdel-Mottaleb. Hierarchical clustering algorithm for fast image retrieval, 1998.

[40] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International Conference on Computer Vision*, pages 2548–2555, Nov 2011.

[41] G.S. Linoff and M.J.A. Berry. *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management*. IT Pro. Wiley, 2011.

[42] Y. Liu, N. V. Chawla, M. P. Harper, E. Shriberg, and A. Stolcke. A study in machine learning from imbalanced data for sentence boundary detection in speech. *Comput Speech Lang*, 20, 2006.

[43] Ying Liu, Dengsheng Zhang, Guojun Lu, and Wei-Ying Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262 – 282, 2007.

[44] Ying Liu, Dengsheng Zhang, Guojun Lu, and Wei ying Ma. Region-based image retrieval with perceptual colors. In *In Proc. of Pacific-Rim Multimedia Conference (PCM2004*, pages 931–938, 2004.

[45] log0. Class imbalance problem. http://www.chioka.in/class-imbalance-problem/, August 2013.

[46] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[47] W. Y. Ma and B. S. Manjunath. Netra: a toolbox for navigating large image databases. In *Image Processing, 1997. Proceedings., International Conference on*, volume 1, pages 568–571 vol.1, Oct 1997.

[48] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, Aug 1996.

[49] B. S. Manjunath, J. R. Ohm, V. V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703–715, Jun 2001.

[50] Stricker Markus, Andreas and Orengo Markus. Similarity of color images. In *Storage and Retrieval for Image and Video Databases III*, volume SPIE Proceedings, Vol. 2420, pages 381–392, 1995.

[51] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 36.1–36.10. BMVA Press, 2002. doi:10.5244/C.16.36.

[52] MathWorks. Image category classification using bag of features. http://-www.mathworks.com/help/vision/examples/image-category-classification-using-bag-of-features.html?refresh=true.

[53] MathWorks. Image classification with bag of visual words. http://-www.mathworks.com/help/vision/ug/image-classification-with-bag-of-visual-words.html?refresh=true.

[54] MathWorks. Image retrieval using customized bag of features. http://-www.mathworks.com/help/vision/examples/image-retrieval-using-customized-bag-of-features.html.

[55] MathWorks. Image retrieval with bag of visual words. http://-www.mathworks.com/help/vision/ug/image-retrieval-with-bag-of-visual-words.html.

[56] Mathworks. Ensemble methods. http://www.mathworks.com/help/stats/ensemble-methods.html, July 2016.

[57] Mathworks. Resume. http://www.mathworks.com/help/stats/classificationensemble.resume.html, August 2016.

[58] Patrick McCurdy. Mediatoil-details. http://mediatoil.ca/Documents/Details/1846, Jan 2016.

[59] Patrick McCurdy. Mediatoil vision. http://mediatoil.ca/Home/About, Jan 2016.

[60] R. Mehrotra and J. E. Gary. Similar-shape retrieval in shape data management. *Computer*, 28(9):57–62, Sep 1995.

[61] S. Mehrotra, Yong Rui, M. Ortega-Binderberger, and T. S. Huang. Supporting content-based queries over images in mars. In *Multimedia Computing and Systems '97. Proceedings., IEEE International Conference on*, pages 632–633, Jun 1997.

[62] V. Mezaris, I. Kompatsiaris, and M. G. Strintzis. An ontology approach to object-based image retrieval. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 2, pages II–511–14 vol.3, Sept 2003.

[63] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, Oct 2005.

[64] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86.

[65] S. Mohanapriya and M. Vadivel. Automatic retrival of mri brain image using multiqueries system. In *Information Communication and Embedded Systems (ICICES), 2013 International Conference on*, pages 1099–1103, Feb 2013.

[66] P. Nemenyi. *Distribution-free Multiple Comparisons*. 1963.

[67] Government of Alberta. About the alberta oil sands. https://www.youtube.com/-watch?v=jcXdLCjzy8s#t=61, Sep 2010.

[68] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, Jul 2002.

[69] M. Kehinde Olorunnimbe, Herna L. Viktor, and Eric Paquet. *Intelligent Adaptive Ensembles for Data Stream Mining: A High Return on Investment Approach*, pages 61–75. Springer International Publishing, Cham, 2016.

[70] Greg Pass, Ramin Zabih, and Justin Miller. Comparing images using color coherence vectors. In *Proceedings of the Fourth ACM International Conference on Multimedia*, MULTIMEDIA '96, pages 65–73, New York, NY, USA, 1996. ACM.

[71] Pembina. Beneath the surface: a review of key facts in the oilsands debate. https://www.pembina.org/reports/beneath-the-surface-oilsands-facts-201301.pdf, July 2016.

[72] A. Pentland, R. W. Picard, and S. Sclaroff. *Multimedia Tools and Applications*, chapter Photobook: Content-Based Manipulation of Image Databases, pages 43–80. Springer US, Boston, MA, 1996.

[73] James Philbin, OndÅ™ej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Minneapolis, MI, June 2007.

[74] Dong ping Tian. A review on image feature extraction and representation techniques. *International Journal of Multimedia and Ubiquitous Engineering*, 8(4):385–395, July 2013.

[75] Richard J. Radke. *Computer Vision for Visual Effects*. Cambridge University Press, New York, NY, USA, 2012.

[76] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*, ECCV'06, pages 430–443, Berlin, Heidelberg, 2006. Springer-Verlag.

[77] Edward Rosten, Gerhard Reitmayr, and Tom Drummond. Real-time video annotations for augmented reality. In *Proceedings of the First International Conference on Advances in Visual Computing*, ISVC'05, pages 294–302, Berlin, Heidelberg, 2005. Springer-Verlag.

[78] Phillipe Salembier and Thomas Sikora. *Introduction to MPEG-7: Multimedia Content Description Interface*. John Wiley & Sons, Inc., New York, NY, USA, 2002.

[79] Eugene Santos Jr. and Qi Gu. Automatic content based image retrieval using semantic analysis. *Journal of Intelligent Information Systems*, 43(2):247–269, 2014.

[80] SeaStar. What are the practical differences when working with colors in a linear vs. a non-linear rgb space? http://stackoverflow.com/questions/12524623/what-are-the-practical-differences-when-working-with-colors-in-a-linear-vs-a-no, November 2012.

[81] Burr Settles. Active learning literature survey. *Computer Sciences Technical Report*, 1648, 2010.

[82] Rui Shi, Huamin Feng, Tat-Seng Chua, and Chin-Hui Lee. *Image and Video Retrieval: Third International Conference, CIVR 2004, Dublin, Ireland, July 21-23, 2004. Proceedings*, chapter An Adaptive Image Content Representation and Segmentation Approach to Automatic Image Annotation, pages 545–554. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[83] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477 vol.2, Oct 2003.

[84] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, Dec 2000.

[85] John R. Smith and Shih-Fu Chang. Visualseek: A fully automated content-based image query system. In *Proceedings of the Fourth ACM International Conference on Multimedia*, MULTIMEDIA '96, pages 87–98, New York, NY, USA, 1996. ACM.

[86] António V. Sousa, Ana Maria Mendonça, and Aurélio Campilho. *The Class Imbalance Problem in TLC Image Classification*, pages 513–523. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[87] Akshay. Sridhar, Scott. Doyle, and Anant. Madabhushi. Content-based image retrieval of digitized histopathology in boosted spectrally embedded spaces. *Journal of Pathology Informatics*, 6(1):41, 2015.

[88] StackExchange. How do decision tree learning algorithms deal with missing values (under the hood). http://stats.stackexchange.com/questions/96025/how-do-decision-tree-learning-algorithms-deal-with-missing-values-under-the-hoo, May 2014.

[89] Peter L. Stanchev, David Green, and Boyan Dimitrov. High level color similarity retrieval. *International Journal ï¿½Information Theories & Applications*, 10:283–287, 2003.

[90] Alexander Statnikov, Constantin F. Aliferis, Douglas P. Hardin, and Isabelle Guyon. *A Gentle Introduction to Support Vector Machines in Biomedicine: Case Studies*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1st edition, 2011.

[91] Renato O. Stehling, Mario A. Nascimento, and Alexandre X. FalcaÌ*f*o. A compact and efficient image retrieval approach based on border/interior pixel

classification. In *In CIKM â€™02: Proceedings of the eleventh international conference on Information and knowledge management*, 2002.

[92] Michael J. Swain and Dana H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[93] H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(6):460–473, June 1978.

[94] H. Tian, Y. Fang, Y. Zhao, W. Lin, R. Ni, and Z. Zhu. Salient region detection by fusing bottom-up and top-down features extracted from a single image. *IEEE Transactions on Image Processing*, 23(10):4389–4398, Oct 2014.

[95] K. Tieu and P. Viola. Boosting image retrieval. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 228–235 vol.1, 2000.

[96] Simon Tong and Edward Chang. Support vector machine active learning for image retrieval. In *Proceedings of the Ninth ACM International Conference on Multimedia*, MULTIMEDIA '01, pages 107–118, New York, NY, USA, 2001. ACM.

[97] Ricardo Da Silva Torres and Alexandre Xavier FalcÃ£o. Content-based image retrieval: Theory and applications. *Revista de InformÃ¡tica TeÃ³rica e Aplicada*, 13:161–185, 2006.

[98] Christopher Town and David Sinclair. Content based image retrieval using semantic visual categories. Technical report, 2001.

[99] UCI. Letter recognition data set. https://archive.ics.uci.edu/ml/datasets/Letter+Recognition, August 2016.

[100] Boyu Wang and Joelle Pineau. Online ensemble learning for imbalanced data streams. *CoRR*, abs/1310.8004, 2013.

[101] J. Wang and X. Su. An improved k-means clustering algorithm. In *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, pages 44–46, May 2011.

[102] J. Wang, M. Xu, H. Wang, and J. Zhang. Classification of imbalanced data by using the smote algorithm and locally linear embedding, 2006.

[103] J. Z. Wang, Jia Li, and G. Wiederhold. Simplicity: semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9):947–963, Sep 2001.

[104] Wikipedia. Absolute color space. https://en.wikipedia.org/wiki/-Color_space#Absolute_color_space, July 2004.

[105] Wikipedia. Lab color space. https://en.wikipedia.org/wiki/Lab_color_space, April 2006.

[106] Wikipedia. Cmyk color model. https://en.wikipedia.org/wiki/-CMYK_color_model, Feb 2015.

[107] Wikipedia. Precision and recall. https://en.wikipedia.org/wiki/-Precision_and_recall, June 2015.

[108] Wikipedia. Chiquita brands international. https://en.wikipedia.org/wiki/Chiquita_Brands_International, July 2016.

[109] Wikipedia. Friedman test. https://en.wikipedia.org/wiki/Friedman_test, May 2016.

[110] Wikipedia. k-nearest neighbors algorithm. https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm, June 2016.

[111] Wikipedia. Support vector machines. https://en.wikipedia.org/wiki/Support_vector_machine, June 2016.

[112] Niraj Juneja William Chen. What are the disadvantages of using a decision tree for classification. https://www.quora.com/What-are-the-disadvantages-of-using-a-decision-tree-for-classification, May 2015.

[113] H. J. Zhang, C. Y. Low, S. W. Smoliar, and J. H. Wu. Video parsing, retrieval and browsing: An integrated and content-based solution. In *Proceedings of the Third ACM International Conference on Multimedia*, MULTIMEDIA '95, pages 15–24, New York, NY, USA, 1995. ACM.

[114] Lei Zhang, Fuzong Lin, and Bo Zhang. Support vector machine learning for image retrieval. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 2, pages 721–724 vol.2, Oct 2001.

[115] Zhi-Hua Zhou. *Ensemble Learning*, pages 270–273. Springer US, Boston, MA, 2009.