Fast Supervised Annotation and Active Learning with Uncertainty for Cloud Mask Dataset Generation

by

Christien Spencer Williams

S.B., Computer Science and Engineering, Massachusetts Institute of Technology (2020)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2022

Author .....
Department of Electrical Engineering and Computer Science
September 7, 2021

Certified by .....
Daniela L. Rus
Erna Viterbi Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Fast Supervised Annotation and Active Learning with Uncertainty for Cloud Mask Dataset Generation

by

Christien Spencer Williams

Submitted to the Department of Electrical Engineering and Computer Science
on September 7, 2021, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Satellite imagery data analysis has made great strides, however still endures inertia due to difficulty generating robust, labeled datasets for complex learners. Variation in data and diverse tasks make it difficult to both generally crowd source to build such datasets, and to offload this responsibility to the small number of expert annotators that exist. Currently, no general machine learning methods can automatically generate data labels in all regimes. A chief data labeling concern for remote sensing projects is cloud mask dataset creation. Using optical satellite images requires detecting accurately all clouds in any image. For many applications, automatic cloud detection methods are not accurate enough. This thesis reformulates the problem away from finding a single automatic algorithm to conduct annotation. We amplify an expert annotator's efforts with an algorithm that learns from his annotations to more efficiently annotate datasets, and an active learning loop that force multiplies this labeling effort. This thesis first contributes a fast, machine learning based annotation system and demonstrates on Sentinel-2 images its efficacy to reach, in four clicks or less, more than 95% accuracy. To obtain these statistics, we constructed an eclectic database of partially cloudy images and its ground truth, and evaluated its accuracy to be greater than 98%. We then show that our fast, supervised annotation is far more accurate than recent sophisticated cloud detectors. Next, we develop an active learning system that employs uncertainty sampling for query selection and uses a modified Efficient Neural Network (ENet) model as its backbone. We evaluate this active learning system by comparing different scoring functions for the uncertainty metric that powers query selection. We show that using this uncertainty measurement, the active learning system performs better using fewer data points. Ultimately, with a minimal number of clicks/annotations, the annotator can build a robust, large, labeled dataset.

Thesis Supervisor: Daniela L. Rus
Title: Erna Viterbi Professor of Electrical Engineering and Computer Science

Acknowledgments

There are many people to whom I feel greatly indebted for helping me successfully complete this thesis. Firstly, I'd like to thank my advisors, Dr. Jean-Michel Morel of the Centre Borelli at École Normale Supérieure Paris-Saclay, and Dr. Daniela Rus of MIT's Distributed Robotics Lab (DRL), for their support and confidence throughout this thesis. A year and a half prior to beginning my thesis work, Jean-Michel and Daniela supported my candidacy as a Fulbright scholar which gave me the opportunity to pursue this study for a year in Paris, France. Additionally, Jean-Michel's guidance throughout my time in Paris was paramount and unwavering.

Secondly, I would like to thank Alexander Amini of the MIT DRL and Dr. Mariano Rodríguez of Centre Borelli. Your consistent and timely responses to my many questions as I navigated my thesis research was benevolent and invaluable to my success. I would also like to thank other members at Centre Borelli, Tristan Dagobert, Carlo de Franchis, Charles Hessel, and Gabriele Facciolo for their insight and aid as well.

Additionally, I would like to thank all of my friends from Fondation des États-Unis and from the basketball courts of Cité Internationale Universitaire de Paris for being my ever present sources of support, laughs, and love during a long year studying away from home.

Last, but not least, I'd like to thank my mother, Adelle, father, Jerome, and brother, Julian. Thank you for your love, encouragement, support, and presence throughout my years of study and through the process of completing this thesis work. Without you, this achievement would not have been possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Satellite imagery products, like Sentinel-2 [17] developed by the European Space Agency, have evolved capabilities for earth observation at high resolution. The Sentinel-2 product provides powerful and accessible spectral images empowering the remote sensing research community. With these vast data available, researchers take preprocessing steps to make data ready for use. Common burdens in the preprocessing stage include image labeling and annotation. Due to the large sizes and visual variance of the images in remote sensing datasets, currently, annotation is not very accurate [68]. Occasionally, groups attempt to generate large, robust datasets via crowd-sourcing annotation work. This can be faulty as sometimes there is poor standardization across annotators. Additionally, some annotators, afraid to give incorrect results, provide labels when they are very confident, but neglect other instances when they are not. Consistency issues also arise in crowd sourced work when annotators are faced with incredibly diverse scenes - for instance, streets look different in Paris, vs. NYC, etc., yet it is desirable for datasets to have diversity in scene colors, object sizes, object shapes, and more. Furthermore, due to data variation and specificity, often only experts are capable of providing reliable annotations in satellite imaging [1].

In attempt to avoid the costs incurred by large, often amateur, human annotator tasks forces, researchers have attempted intelligent strategies based on statistics and machine learning. Studies show, however, that there are no fixed machine learning

[1]Personal communication from satellite imaging experts at Centre Borelli and Kayrros, Inc.

methods that can work in a multitude of cases to automatically annotate the data; data varies greatly and tasks are vastly different. For example, some works have investigated the use of unsupervised, clustering-based methods that avoid any human intervention [9, 67]. This strategy struggles, however, in that it relies on strong measures of similarity between data which may not present itself as strongly in real problems [68]. Other methods have instead attempted supervised classification to achieve better classification performance [2, 20, 32, 53]. Fully convolutional networks (FCN) elicit optimism due to their ability to extract dense features, to assimilate context [5], and to operate on inputs of arbitrary sizes [29]. Nevertheless, these fully-supervised strategies often require vast labeled datasets to learn from, thus bringing the problem back full circle to employing large human-annotator task forces to prepare the necessary data. Additionally, most annotation systems for deep learning and imaging use bounding boxes; this is not optimal for making precise masks; thus, one needs pixel-by-pixel annotations, which again necessitates human annotation intervention. Ultimately, it seems this problem requires: 1) some human intervention from expert annotators to tackle the problem of consistent, standardized datasets for niche and diverse data, and 2) intelligent machine learning tools to force multiply the annotator's work to tackle the overwhelming task of labeling the vast data. This thesis seeks to find a solution to this problem by answering the question: can you drastically minimize the number of clicks an expert annotator makes to increase the speed of accurate and thorough satellite image annotations?

This study will focus on the creation of cloud mask datasets, i.e., datasets consisting of pixel-wise annotation bitmaps (corresponding to input images) that classify pixels as cloud or non-cloud. In a majority of remote sensing analysis tasks, cloud detection is a burden and a preliminary step to discriminating desired content in images. Tasks like atmospheric correction, land cover classification, change detection, or inversion of biophysical variables first require accurate cloud detection/segmentation. While seemingly a simple task, cloud detection is difficult over land because when clouds are significantly larger than pixel size, it's difficult to distinguish them from background objects [31]. The vast shapes, sizes, and colors of clouds and earth surface

landscapes can be misleading, bright landscapes can be easily confused for clouds, and additionally, semi-transparent clouds' reflectance often resembles both cloud-like and land-like signals [7]. Convolutional neural networks show promise in achieving high accuracy cloud detection; however, they require cloud mask datasets. Currently, there is a lack of high-quality labeled datasets for many of the well-known satellite products (i.e., Sentinel-2, Landsat 8, etc.). Thus, cloud mask dataset creation for remote sensing images is a compelling domain to test our primary question. Ideally, with only a small handful of annotations per image, and by only annotating a small sample of an entire unlabeled dataset, an expert annotator endowed with machine/algorithmic aid can generate a robust image-cloud mask dataset.

This thesis' work proposes a two-level system that includes a fast annotation process for expert annotators of a small dataset, and a larger learning loop that will subsequently learn from these data to annotate a greater corpus of data. By breaking the system down like this, we are able to 1) focus an expert annotator's attention on a small subset of unique, informative instances, thus maximizing the efficiency of annotator resources; 2) endow the annotator with a fast, efficient pixel classifier that enables them to generate a segmentation for an entire image while only labeling a few pixels; 3) use a separate model to extrapolate knowledge learned from annotators to classify the larger, comprehensive dataset, and to identify further uncertain instances to be labeled by the annotator to improve the segmentation accuracy of the dataset. Comprehensively, with this two-level learning loop, one will be able to train a model that can perform automatic cloud detection and annotation analysis.

1.1 Contributions

In summary, the contributions of this thesis will be as follows:

1. A fast supervised cloud mask generation process for Sentinel-2 images that allows expert annotators to make preliminary annotations and then iterate, adding subsequent annotations until achieving the desired cloud mask;

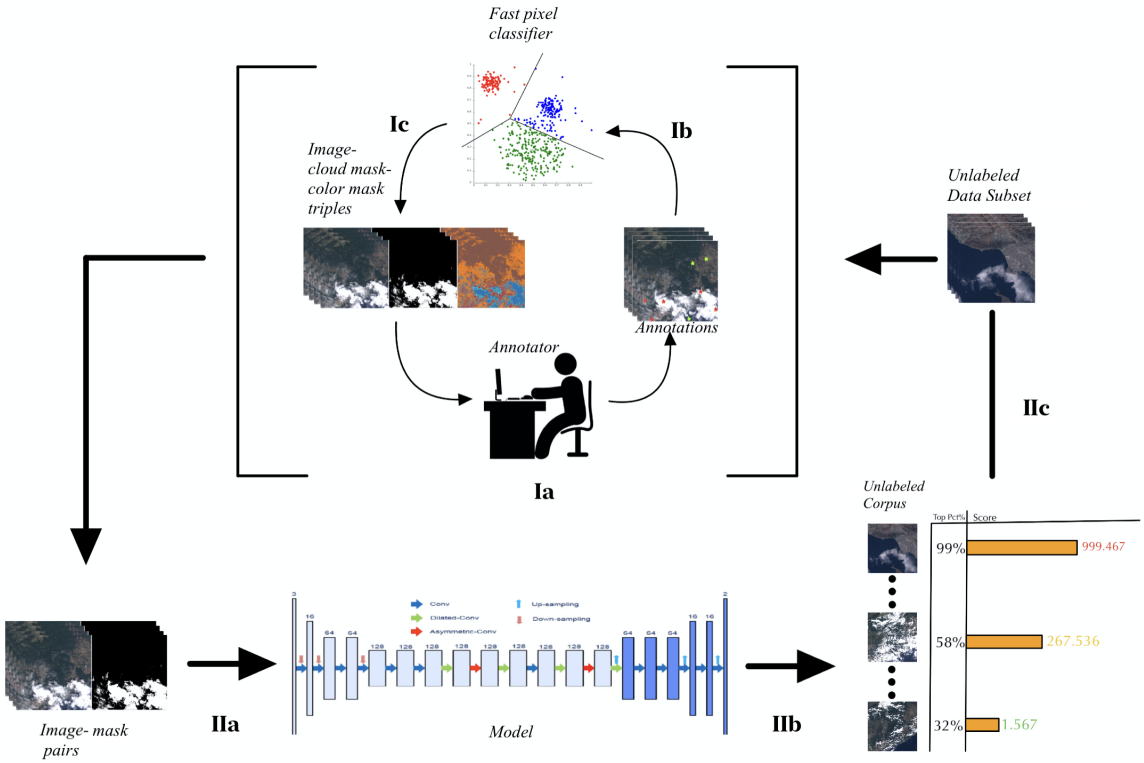


Figure 1-1: **Architecture of the Two-Level Active Learning Annotation System**. Beginning with **Level I**, in step **Ia**, the oracle begins annotating a subset of data. In step **Ib**, the oracle's annotations are passed to a fast pixel classifier as training data. The classifier then classifies all pixels in the input images, producing cloud masks and color masks during **Ic**. Evaluating this output visually, the oracle can make further annotations. This loop continues until the oracle is satisfied with the outputted cloud masks. Once satisfied, **Level II** begins as the image-mask pairs from **Level I** are used as input training data for ENet (step **IIa**). Subsequently, in step **IIb**, the trained ENet model is run on the unlabeled data pool; each instance is given an informativeness (uncertainty) score. This score is used as the metric for the active learning query step. Seen in step **IIc**, images with scores in the top 10th percentile are queried to the oracle for annotation, and a new system cycle begins.

2. A modified ENet (efficient neural network) architecture that integrates uncertainty estimation that provides a measure of uncertainty with the output which guides the subsequent annotation steps (ultimately minimizing the total annotation necessary);

3. An eclectic dataset of 1830x1830x15 (HxWxC) partially cloudy images and corresponding cloud masks with 98% cross-validated accuracy;

4. Experimental validation of the fast mask generation process compared to automatic detector algorithms; and

5. Experimental validation of an active learning system, employing the modified ENet model and uncertainty score metrics, used to produce a dataset of cloud mask images with high segmentation accuracy.

1.2 Organization

This thesis is organized as follows. Chapter 2 highlights the related work. Chapter 3 presents the proposed annotation process. Chapter 4 introduces Bayesian deep learning, inference and uncertainty, the relationship with active learning acquisition functions, and the neural network architecture. Chapter 5 discusses the satellite used, highlights brief examination that helped clarify and guide subsequent experimental focus, and then outlines the experimental pipelines. Chapter 6 discusses experimental results. Lastly, in Chapter 7, this thesis summarizes this thesis' work and capture future work.

Chapter 2

Related Work

2.1 Existing Annotated Cloud Datasets

There is a deficiency in comprehensive, high quality, labeled, cloud datasets. While some datasets exist, they each have weaknesses. The 2016 Sentinel-2 Hollstein *et al.* dataset [37] yields annotated polygons in Sentinel-2 images acquired over several different sites at various dates. The polygonal cloud masks are necessarily imprecise on the often-ragged clouds. CloudNet [46] contains 120 small images of size 224×224 pixels, all coming from a single Sentinel-2 acquisition. The cloud mask was drawn by hand (using Adobe Photoshop and ENVI), on the True Color Image (RGB) band of the L1C products. While the quality of this manual segmentation is good, the image contains a lack of diversity in weather conditions, seasons, and geographies, as all images come from a single Sentinel-2 capture.

A few datasets prove to be exceptions as they are of respectable quality. A dataset produced in 2019 by Baetens, Desjardin and Hagolle [8], provides reference masks for 38 Sentinel-2 scenes [12]. The dataset has six classes: low clouds, high clouds, cloud shadows, land, water, snow. All pixels are annotated with a resolution of 60m; upon visual inspection it appears to be of high quality. This dataset, however, lacks scene coverage in many regions of the world. Thus, for analysis in those specific regions with unique land cover and spectral, atmospheric conditions, researchers may be interested in generating their own datasets. Conversely, consisting of 96 scenes,

the Landsat 8 Cloud Cover Assessment Validation dataset, has more diversity [18]. Nevertheless, this dataset consists of frames obtained by Landsat 8 only, and as new satellite products are released with a wider variety of bands and other spectral data, new datasets will be desired from these more modern products.

2.2 Automatic Cloud Detectors

One approach to generate the desired cloud mask datasets could be to use existing automatic cloud detectors. Several works attempt to use physics or statistics based approaches instead of supervised learning approaches which require the desired labeled dataset to begin with. In [13], Dagobert *et al.* proposed an automatic cloud detector based on the spectral parallax of pushbroom satellites such as Sentinel-2, Landsat-8, Pléiades or RapidEye. They employ a region growing algorithm followed by a statistical validation of regions with coherent parallax. Shin and Pollard use Along-Track Scanning Radiometer (ATSR) in an attempt to detect clouds over seas [64]. In a different vein, Manizade *et al.* [52] apply discrete correlation to binary series obtained by the slicing of the 8-12 μm infrared band into 18 temperature ranges. This enables the refinement of the cloud altitude estimation up to ±390 m. In [69], researchers develop the FMask algorithm for Landsat 5, 7, and 8. FMask uses surface reflectance and the brightness temperatures of the Thermal Infra-Red (TIR) Channels to detect cloudy pixels. Developed by the European Space Agency, Sen2core is a program that processes Sentinel-2 level 2A product data. Like FMask, Sen2core is a thresholding algorithm that takes into account reflectance levels and various band ratios such as infra-red/green or near-infra-red/sand. Unlike the FMask and Sen2core algorithms which are mono-temporal methods that process only a single image to make detections, MAJA [31] is a method that operates on multi-temporal data and combines the Multi-sensor Atmospheric Correction and Cloud Screening (MACCS) and the Atmospheric and Topographic Correction methods. It is used by the French Space Agency [7]. FMask, MAJA, and Sen2core are some of the most well-known and used methods for achieving cloud masks, however neither is ubiquitously suc-

cessful. Thus, users must weigh the benefits of each and scrutinize their desired use case before using them. For example, Sen2Core has shown significant underestimation in dense cloudy/shadowy areas [50]. S2cloudless is a machine learning based detector that uses tree-based learning algorithms with XGBoost and LightGBM [70]. Showing promising results and being well accepted in research communities (as it has been downloaded 47,000 times [58]), S2cloudless has also shown weaknesses as it has struggled to perform well in difficult cloud covers such as those in the Amazon rain forest [60]. In sum, we've seen that existing cloud detectors are not yet performant enough to rely solely on them for the generation of the desired cloud mask datasets. Evidently, you stumble upon a chicken and egg problem.

2.3 Cloud Annotation With A Human in the Loop

A more recent study attempts to use machine learning with a human in the loop to build a labeled cloud dataset. In their work, Baetens *et al.* [7] attempt to reformulate the focus away from an automatic detector towards a fast annotation system for generating cloud mask datasets. They use an active learning process, called Active Learning Cloud Detection (ALCD), with a random forest learner to learn from annotations and reflect back to the annotators the most uncertain points to annotate next. The paper proposes making annotations via clicks, instead of polygons. They further recommend using a dilation kernel for helping to classify fuzzy edges. Both of these strategies will be employed in our fast annotation process. ALCD, struggles, however, in that it requires a completely non-cloudy reference image for its analysis to be successful. This, thus, limits its usefulness in permanently cloud regions, such as Guyana or Congo [7].

2.4 Active Learning

As stated earlier, this thesis seeks to develop a two-level system that grants the ability to train a model that can perform automatic cloud detection and annotation analysis,

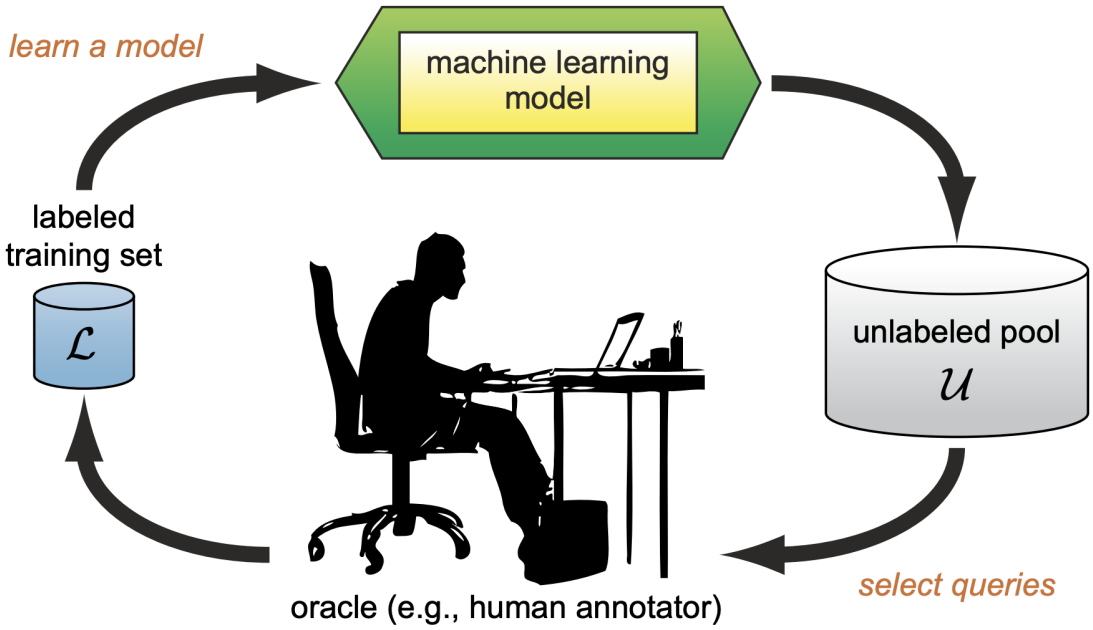


Figure 2-1: Illustration of the pool-based active learning process [63]

enabling swift dataset creation. Centrally, the goal is to force multiply the efforts of a single expert annotator. One strategy to achieve this is to build a system to direct an oracle/annotator to the essential instances to label in an unlabeled dataset such that, after only labeling some of these maximally informative instances, the model can learn from these annotations and perform approximately as well as if it had learned from annotations from every instance in the dataset. This goal is precisely the focus of a methodology referred to as active learning. Using active learning, an oracle will only have to annotate a small portion of the data; this contributes to minimizing the total clicks/annotations the oracle must make. The arena of active learning has substantial literature elaborating on its formulation, theory, and applications.

The primary notion of active learning is that a model can achieve better accuracy, learning on fewer instances, if it is able to choose what data it learns from. The active learning process is visualized in Figure: 2-1. The learning system begins with a relatively small set of labeled training data. The machine learning model learns from these data, and subsequently evaluates the instances in a larger unlabeled pool of data. The active learning system then makes queries to an oracle, inquiring about

the labels of instances which have been selected as a result of some "query framework"/"activation function". After the system makes a query to the oracle, there are no further assumptions made by the learner about the instance or its significance in the dataset [63]. Once the annotator annotates the instance, it is added to the set of labeled training data.

2.4.1 Query Scenarios

In this section, this thesis discusses the three main query scenarios as illustrated in 2-2.

Membership Query Synthesis

In membership query synthesis, the learner may pose queries on any instance in the unlabeled instance input space; this includes instances that the learner generates de novo [63]. This method is tractable and computationally efficient in problems with infinite domains [4]. Queries generated de novo, however can cause problems. For example, in [38] the methodology resulted in the learner generating de novo hand written characters that were ambiguous symbols that made no sense to human annotators. Later, however, in [42], the authors generate promising results for the membership query synthesis methodology, employing a "robot scientist" that executes the generated de novo queries (which were biological experiments), by actually running the experiments.

Stream-based Selective Sampling

In stream-based selective sampling, one assumes that the process of soliciting an instance for querying is free or cheap. As a result, one can actually sample the instances from a distribution; then a learner decides whether or not to make a query of the instance to the oracle. In the case that the distribution is uniform, stream-based selective sampling behaves like the membership query synthesis strategy. On the other hand, if the distribution is not uniform, with stream-based selective sampling,

sampled instances are guaranteed to be sensible [63].

Pool-based Sampling

Pool-based sampling, visualized in 2-1, is inspired by the notion that one can collect large pools of unlabeled data at one time. Then, an active learning system can select instances of interest from said pool (often assumed to be static/non-changing). Instances may be selected in a greedy fashion or due to some acquisition function that reflects an information measure on the instance. Pool-based sampling is our area of focus [63].

2.4.2 Query Framework/Acquisition Functions

Query frameworks/acquisition functions are the methods of cleverly selecting instances for querying in pool-based sampling. Ultimately, given a machine learning model M that makes predictions as a function of some input data, a data pool D_{pool}, and unlabeled inputs $x \in D_{pool}$, an acquisition function $a(x, M)$ is a function of x that the active learning system uses to decide where to query next, by finding

$$x^* = argmax_{x \epsilon D_{pool}} a(x, M).$$

I explore several methods to serve as this acquisition function, $a(x, M)$. One category of methodologies is a pure scoring/uncertainty based methodology. One may select instances for which the posterior distribution evaluated at that point has the most entropy. One could also attempt a method referred to as variation-ratios. Here, one selects the instance whose classification with highest probability is smallest. Both strategies evaluate lack of confidence [23]. The equations for both (hereon referred to as entropy formula and variation ratios formula, respectively) are

$$\mathbb{H}[y|\boldsymbol{x}, D_{train}] := -\sum_c p(y = c|\boldsymbol{x}, D_{train}) \log p(y = c|\boldsymbol{x}, D_{train}),$$

$$\text{variation-ratio}[x] := 1 - \max_y p(y|x, D_{train}),$$

where $\boldsymbol{x}$ is an unlabeled input, and y is the predicted classification that takes on value c ("cloud", or "non-cloud").

Another category of query framework explored in this thesis takes a diversity-based sampling, core-set approach. Ultimately with this approach, one aims to find a minimum subset of a larger labeled dataset such that when the active learning system learns using this smaller subset, it is "competitive over the whole dataset" [61].

The active learning work discussed thus far will be applicable for selecting image crops from a large dataset of satellite imaging data to be annotated by an oracle. However, active learning has also been used at a lower layer in the satellite imaging annotation process. In [7], the same study as referenced in 2.3, the authors recognize the necessity of thorough datasets of image-mask pairs for improving detection capabilities in satellite imaging. Their analysis uses the Sentinel-2 Level 2A product developed by the European Space Agency. The product provides users with surface reflectance measurements and a cloud/cloud shadow mask. In [7], the authors develop a program called Active Learning Cloud Detection (ALCD) to generate cloud masks to validate the Sentinel-2 product output. Using the ALCD, an oracle (human annotator) can label a few points via clicking on an image. Subsequently, a random forest model is learned which produces a classification on the input image generating a cloud mask. The oracle visually evaluates incorrect and uncertain pixels in the mask and corrects them with further annotations. This annotation/learning loop continues until the oracle is satisfied with the mask.

2.4.3 Further Works

Novel active learning methods have surfaced in regimes well outside of cloud segmentation. In [25], the authors develop a methodology they refer to as the “farthest-first compression” (FF-Comp). With FFC, for each active learning step, the authors use a model-based, core-set technique to compress the dataset (inputs) based on embeddings derived from neural network activations. Their work performs better than passive learning on MNIST, CIFAR-10, and CIFAR-100. In [24], the authors develop a novel active learning strategy that uses Bayesian deep learning and an information

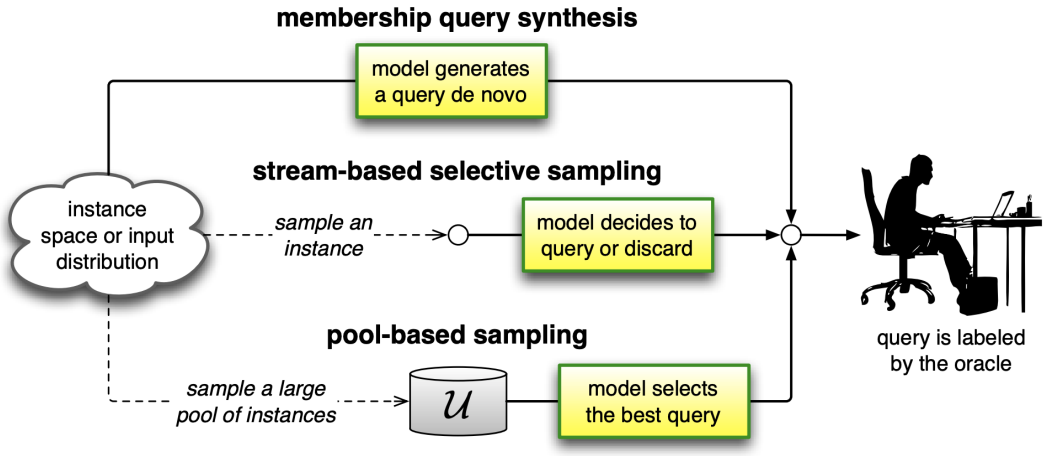


Figure 2-2: The three main active learning scenarios [63]

measure (BALD) that maximizes mutual information between network predictions and the model posterior. The BALD metric enables the active learning system to effectively select the instance with the largest variance going into the softmax activation layer. The authors show that BALD outperforms random sampling active learning and semi-supervised techniques on the MNIST dataset. BALD also outperforms a uniform acquisition strategy when applied to the ISIC 2016 melanoma diagnosis dataset. Furthermore, BALD performs better than a network trained on the entire dataset. This is hypothesized to be because BALD avoids selecting noisy datapoints with high aleatoric uncertainty (discussed in more detail in 4). The authors in [43], propose Learning Active Learning (LAL) methodologies. They show that when the classes in the dataset are unbalanced, using entropy as the active learning acquisition metric is sub-optimal. The paper acknowledges there are many complex factors that affect the class distribution, and thus they use properties of the data and the classifiers to estimate potential error reduction. They employ a regression model that operates on a manifold characterized by parameters of the classifier and the datapoints, and adapts query selection by annotating a selected datapoint in an informative classifier state. The LAL methodologies succeed in a variety of regimes, including synthetic data, biomedical imaging, economics, molecular biology and high energy physics. Finally,

in the approach presented in [10], the authors reformulate the active learning problem as one of learning in an adversarial environment. Their work strives to improve the system's robustness to outliers and variance due to adverse model instantiation that greedy selection sometimes lacks. Their innovation is to stochastically select points for labeling, where the probability simplex that defines said chance of selecting a point, i, depends on a bounded, expected regret and the most recently observed loss. Named, AdaProd+, this innovation outperforms uniform, entropy, and uncertainty sampling on the ImageNet, SVHN, and CIFAR10 datasets. Ultimately, this paper's contribution demonstrated low regret on predictable instances, and was robust to adversarial ones.

2.5 Uncertainty Estimation

In attempt to achieve our annotation goal, we opted to exploit a model's understanding of its uncertainty in predictions to best inform an active learning process. Uncertainty estimation is critical in machine learning systems. Often, outputs are extracted from machine learning models and assumed accurate; however these outputs may not be [44]. For example, in a fatal 2016 incident, an assisted driving platform's perception system mistook a white side of a trailer for a bright sky [1]. In another account, a perception system identified two African Americans as gorillas [30].

Uncertainty metrics often concern themselves with calibration and generalization to domain shift. Calibration is a notion that evaluates divergence between subjective forecasts and long-term horizon frequencies [15, 16]. Generalization to domain shift (aka "out-of-distribution examples") implies a model can measure what it knows or does not know [35]. A lot of work has been done to enable neural networks (NN) with uncertainty estimation capabilities. Much of this work is in the Bayesian regime, wherein a prior is placed over the weights of the model, and after training, the weight posterior distribution is computed to measure predictive uncertainty. Intractability in marginalization computation has led to several approximations, such as the Laplace approximation [49], Markov chain Monte Carlo (MCMC) methods [54], and varia-

tional methods [27, 48]. Bayesian NNs are often difficult to implement, however. In [22], the authors formalize Dropout as Bayesian approximation, leading to the use of Dropout for practical Bayesian NN implementation. Researchers have also tried sampling across an ensemble of networks to produce uncertainty measurements, however this methodology demands great computational resources and can be difficult to deploy and parallelize in real-time. In the end, the information gained from uncertainty estimation will be the metric the active learning loop uses to inform its efficient sampling/querying.

2.6 Segmentation and Mask Prediction

The nature of analysis task that the model in the active learning framework is faced with, is one of segmentation, and ultimately mask prediction: the model is learning a mapping between an input image and a cloud mask of that image. Several works have developed novel network architectures that have pushed the state-of-the-art in this space. In [47], the authors pioneer use of fully convolutional networks (FCNs) for end-to-end training in image segmentation tasks. FCNs begin with an image of arbitrary size and output a segmented image with the same size. U-Net, originally designed for biomedical image segmentation, extended the FCN concept. U-Net is composed of two parts: one that downsamples the input image down to features, and another that upsamples up to the segmented image [59]. In [45], researchers develop the Feature Pyramid Network. This architecture combines a top-down and bottom-up pathway to join low and high resolution features. The DeepLab architectures expanded upon this and employed atrous convolution to attain multi-scale context [11]. In general, each of these architectures attempt to capture information about global visual context to improve segmentation predictions. The state-of-the-art networks cleverly link different regions of the image to learn relationships between the various objects in the image [55].

Chapter 3

Fast Supervised Cloud Mask Generation

In this chapter, this thesis presents the first level of the annotation system. This level consists of a fast supervised cloud mask generation process for Sentinel-2 images. First an annotator selects cloudy pixels by clicking on the image (via an editing application - for this thesis, we use QGIS). Then, a fast pixel classifier learns from these data and classifies the rest of the image. Finally the annotator completes the segmentation via a few further clicks. We employ this process first, without limitation on the number of clicks, on an eclectic set of partially cloudy images, to generate ground truths with 98% cross-validated accuracy.

3.1 Pixel Classifiers

The choice of pixel classifiers was the chief question at hand. Four different methods are used to classify pixels: a support vector machine (SVM), a random forest, a mixture model with pixels as inputs, and a mixture model with a patch of pixels as inputs. Clicks from QGIS that annotate "cloud" or "background" were first dilated by 2 meters in radius (inspired by [7]). Subsequently, the annotations and expansions were fed to the learner as supervised training data. For each classification method, the following cycle was carried out after training:

1. The classifier classifies all pixels of the input image.

2. An output mask is generated from this classification.

3. Further annotation corrects the visible errors in the output.

4. The model is trained again on the annotations.

This process continues until the annotator is satisfied with the generated output mask.

3.1.1 Support Vector Machine

A support vector machine (SVM) is a linear classifier. Using it for image segmentation begets finding a margin to separate pixels into classes in color space (in the dimension of the number of image channels used). For the support vector machine annotation (SVMA) methodology, the 15-channel input pixels are first transformed using a radial basis function (RBF) kernel with kernel coefficient parameter, gamma, set to 1.0. By mapping the input vector into a higher dimensional feature space, the RBF enhances the ability to find a separating hyper-plane. The 15 channels include all 13 spectral bands innately provided by the Sentinel-2 product in addition to the derivative Normalized Difference Vegetation Index (NDVI) and Normalized Difference Water Index (NDWI) indices. The SVM is then trained on these transformed data points.

3.1.2 Random Forest

A Random Forest (RF) machine learning method is a decision tree model often used when the classes of a dataset are unbalanced, is high dimensional data, or contains outliers [41]. For the random forest annotation methodology (RFA) in this thesis, each input pixel consists of all 15 channels. As with SVMA, after training on the annotated data, the cycle above (3.1) was carried out.

3.1.3 Bayesian Mixture Model

Mixture models have the ability to cluster pixels in (potentially high dimensional) color space. We use a Bayesian mixture model (BMM) with a Dirichlet distribution prior to infer an approximate posterior distribution over the parameters of a mixture model. Using this non-parametric model, the number of parameters (pixel clusters) is inferred from the data, thus one need not guess nor choose this value via extensive experimentation. The model is first trained to create pixel clusters. Then, all the pixels annotated as cloud/non-cloud ared used to "vote" on the class of the cluster they belonged to. The majority vote becomes the class that the cluster is assigned to. By design, any pixel cluster from the mixture model that is not voted in the "cloud" class is designated as "background." Like with SVMA and RFA, after training, the cycle above is carried out to complete the Bayesian Mixture Model annotation (BMMA) process.

3.1.4 Bayesian Mixture Model with Patches

We expand upon the BMM concept to provide textural information to the learner. We implement BMMA for pixel patches wherein each click in QGIS is expanded to a $n \times n$ patch ($n \in \{3, 5, 7\}$). For the original BMMA, each click is dilated and the dilated pixels are separate observations. Now, however, this dilation is unraveled so that each expansion around a click (let us say the click was expanded to a 3x3 dilated patch), turns into a single feature of size $9 * num\_channels$. PCA is then run on each patch (keeping the components comprising 98% of the variance) prior to the patch being used as the feature vector input. Subsequently, the same cycle above (3.1) is run to round out this process.

3.2 Ground Truth Cloud Mask Creation

To compare the discussed methods, we needed accurate ground truths. This thesis contributes a dataset of 1830x1830x15 (HxWxC) images and corresponding cloud

masks. The first experiment's dataset includes 12 masks and associated images: 4 scenes with 5, 1, 4, and 2 time-series images respectively. The second and third experiment share a dataset of 12 masks and associated images: 12 separate scenes captured on arbitrary dates. We annotate using QGIS, a geographic information system (QGIS) software. Only "cloud" and "background" annotations are made. Two methods are used for generating ground truth. Following guidance from [6], 15 spectral bands are used (or at least considered).

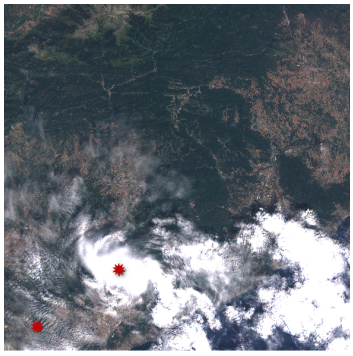
3.2.1 Creating a First Ground Truth by Random Forest

A Random Forest is employed using all 15 channels. First, pixels selected by clicks in the QGIS software and annotated "cloud" or "background" are dilated by 100 pixels (2 meters in radius; inspired by [6]). This is seen in **3-1a**. All these annotated points and their dilations are used as training data. After training, we employ the model to classify all pixels in the input image and generate a cloud mask from the pixel predictions. Cross-examining the RGB image and the mask "in progress", the annotator makes further annotations. This process continues until reaching the most accurate cloud mask.

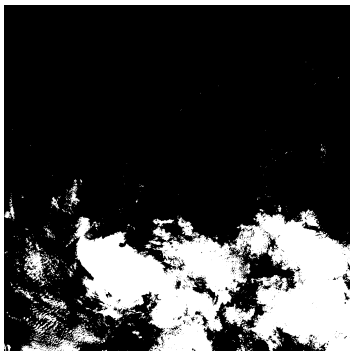
An optional median blur or a morphological opening with kernel radius 2 are used to smooth out imperfections in the image. The procedure to build a ground truth using the RF methodology is performed twice by two different expert annotators for cross-validation. The percentage error between two applications of this same annotation process is computed as a pixel-by-pixel comparison made across the two instances of this same ground truth generating process. The error is $\frac{n_{incorrect}}{N_{totalpixels}}$ (reflected in **Table 3.1**).

3.2.2 Creating a Second Ground Truth by Bayesian Mixture Model

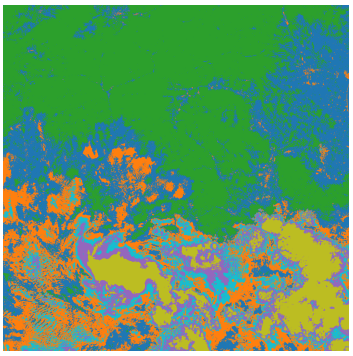
A BMM is used to cluster pixels in color space, however not all 15 channels are used. It was discovered that BMM using only Sentinel-2 band, B1, often provided the best



(a) Original image with two "cloud" clicks.



(b) The BMMA output cloud mask.



(c) The BMMA output color mask.

Figure 3-1: First the annotator makes annotations, as in (a). After running the BMM method, we output the cloud mask and color mask [(b) and (c)].

output mask. Other times, the superior band(s) to use was only B11, or bands B1 and B11 together, or the RGB bands. We tested each of these band combinations to construct the most accurate ground truth.

Color masks are used to aid the annotator. A color mask is generated by painting every pixel in the input image that has been grouped into the same class (by the mixture model) the same color. With this, the annotator can visualize how pixels are clustered, see which clusters belong to clouds, and annotate accordingly. **Figure 3-1** illustrates the process. After the annotators make annotations, **3-1a**, they run the BMM method, resulting in the cloud mask and color mask [**3-1b** and **3-1c**]. Seeing the orange region in the color mask, a cluster of pixels that does not yet have a "cloud" annotation as it should, the annotator can subsequently annotate one of the pixels in that region in **3-1a**.

Similar to that mentioned in 3.2.1, an optional median blur or a morphological opening with kernel radius 2 is used. A percentage error between two applications of this same annotation process is also computed as a pixel-by-pixel comparison made across two instances of this same BMM ground truth generating process.

We additionally compute the relative error across the two ground truth generation methods. **Table 3.1** reports first the internal relative error between generated ground truths from two applications of BMM and RF, respectively. We next report the external error between the ground truths generated by both methods. Since two

	Error Percentage
BMM internal	1.80%
RF internal	3.30%
BMM-RF external	4.10%

Table 3.1: Internal relative error between the generated ground truths from two applications of the BMM and RF methods, respectively. External error between the ground truths generated by the BMM and RF methods.

applications of BMM and RF generated two ground truth images each, computing the error between all combinations of the outputs yielded four error percentages. We report the average error percentage from these results.

3.3 Procedure Overview

In summary, we present the procedure for the fast supervised cloud annotation step. After being presented with a dataset of unlabeled images, the oracle clicks annotations, labeling representative pixels as cloud or non-cloud. The classifier then learns from the annotations. For RFA and SVMA, the procedure solely looks at the annotated pixels during this training step. Subsequently, for RFA and SVMA, the classifier classifies every pixel in the input image, generating a mask that segments the image into cloud and non-cloud partitions. For BMMA, the procedure first clusters all pixels in the input image. Then, the annotated pixels "vote" on the class of the cluster they belong to, with the majority vote becoming the class the cluster is assigned to. Next, for BMMA, the procedure assigns every pixel the same class label as that of the cluster that the pixel belongs to, generating a cloud mask. A color mask is also made during the BMMA procedure. This is generated by taking all pixels in the input image that have been grouped into the same class, and painting them the same color. Finally, for each of RFA, SVMA, and BMMA, the annotator evaluates the mask(s) visually to decide if further annotations should be made to improve the segmentation accuracy.

Algorithm 1: The Fast Supervised Cloud Annotation Step

Input: A dataset, D, of size n, of unlabeled data provided to the oracle for annotation.

while *Image masks have not been made or require further modification* **do**

- Annotation
- Fast pixel classifier training
- Whole image classification
- Visual evaluation

Chapter 4

Uncertainty of Cloud Classification: The Active learning Step

The previous chapter discussed the use of supervised machine learning techniques, like mixture models, random forests, and support vector machines to classify pixels within an image. This chapter will step to a higher level in the annotation system; instead of learning on a pixel level, learning occurs on the image level. Stepping into this arena, we begin employing use of deep learning, artificial neural networks. We use convolutional neural networks to learn a task that maps an input image to a segmentation image that classifies all of the input pixels as either cloud or non-cloud. While it can be effective, blindly taking the output mapping of a deterministic neural network does not make use of all information that the network can provide. One crucial datapoint lacking here is the answer to the question, "am I [the model] uncertain about this output?" It is with this information that the model can more cleverly exploit the data it's learning from to successfully generalize and achieve in its segmentation task. Ultimately, it is with this information, that the active learning component of the system prospers.

The goal of this thesis is to minimize the number of clicks an expert annotator makes to generate a robust dataset. Following the procedures of active learning, the system begins by learning on only a small portion of the dataset. It then uses its knowledge to evaluate its confidence segmenting new, unlabeled images. Sub-

sequently, the system queries labels from an oracle on the most uncertain of these unlabeled images. In [26], the authors show that iterating with such an active learning loop, in lieu of initially training on the entire dataset, decreases the total number of annotations for performance convergence. The system finds the most informative subset of data that best represents requisite information for comprehending the entire dataset. As neural networks' failure modes are commonly out-of-distribution domains, their predictive confidence provides a window into which instances will best inform future decisions in these domains. Uncertainty quantification can help facilitate this when this measure itself is the score employed by active learning to notify new queries [65].

There are two types of uncertainty one may want to model: aleatoric uncertainty, and epistemic uncertainty. Aleatoric uncertainty pertains to noise intrinsic to the dataset. For instance, a broken pressure sensor whose recordings will naturally include some random variation due to the malfunctioning recording mechanism. No matter how much additional data is recorded, this uncertainty/error won't be able to be reduced. Epistemic uncertainty pertains to uncertainty in model parameters, revealing unawareness of which model produced the data [40]. Epistemic uncertainty can be unlocked by leveraging Bayesian Deep Learning, and will be used to inform our active learning system. This chapter will present a modified ENet and uncertainty scoring functions which power the active learning loop.

4.1 Bayesian Deep Learning

4.1.1 Problem Formulation

The objective of Bayesian neural networks is to learn a posterior distribution over the model's weights, given the data and observations: $P(\mathbf{W}|\mathbf{X}, \mathbf{Y})$. This posterior represents possible model parameters given the data and observations [40]. Using the posterior and its moments, one can capture uncertainty. The posterior can be reformulated using Bayes Rule,

$$P(\mathbf{W}|\mathbf{X},\mathbf{Y}) = \frac{P(\mathbf{Y}|\mathbf{X},\mathbf{W})P(\mathbf{W})}{P(\mathbf{Y}|\mathbf{X})}.$$

While this is simple to formulate, this is often intractable to compute in practice as the $P(\mathbf{Y}|\mathbf{X})$ term cannot be evaluated analytically. Researchers developed variational inference (VI) methods to address this [27]. In VI, one uses a simpler distribution, $q(\mathbf{W})$, as an approximation to achieve the predictive distribution,

$$q(\mathbf{Y}|\mathbf{X}) = \int P(\mathbf{Y}|\mathbf{X},\mathbf{W})q(\mathbf{W})d\mathbf{W}.$$

In many instances, $q(\boldsymbol{W})$ is cleverly defined to be a part of the family of mean field approximations, wherein there are no dependencies between the variables. $q(\boldsymbol{W})$ can thus be easily factorized, and these marginals are evaluated by minimizing the Kullback–Leibler divergence with the posterior distribution. VI methods can still suffer from computational costs [36, 56]. Dropout is an efficient method for complex models to approximate VI.

Dropout was initially developed to improve neural networks' generalization capabilities. One would place independent, identically distributed Bernoulli random variables upon a network's neurons which would govern, with some probability, if the neuron was left "on" or "off" during training [21]. Consider that at each step, employing dropout is to sample a Bernoulli random variable for each weight, distributed as

$$\mathbf{z}_t^{(w)} \sim Bernoulli(p) \; \forall \; w \; \epsilon \; \mathbf{W}.$$

When we multiply this by the networks' nodes, it provides a stochastic sample of the weights, as

$$\mathbf{W}_t = \{\mathbf{z}_t^{(w)} \cdot w\}_{w\epsilon\mathbf{W}}.$$

In [22], the authors show that using dropout is equivalent to sampling weights that achieve realisations from the Bernoulli distribution. They then show that using the dropout approximation is equivalent to minimize the Kullback-Liebler divergence between the approximation posterior and the true posterior. As a result, the predictive posterior is determined to be

$$p(y^*|x^*, \mathbf{X},\ \mathbf{Y}) \approx \int\ p(y^*|x^*, \mathbf{w})q(\mathbf{W})d\mathbf{w} \approx \frac{1}{T}\sum_{t=1}^{T} p(y^*|x^*, \hat{\mathbf{w}}_t),$$

where the likelihood $p(y^*|x^*, \hat{\mathbf{w}}_t) = Categorical\left(\ exp(\hat{\boldsymbol{f}})/\sum_{d'} exp(\hat{\boldsymbol{f}}_{\boldsymbol{d'}})\right)$, $\hat{\boldsymbol{f}} = \hat{\boldsymbol{f}}(x, (W_i)_{i=1}^{L})$ as the random output of a Bayesian neural network, and T is the number of stochastic runs through the network. It is this predictive posterior that grants the ability to compute metrics for uncertainty.

4.1.2 Inference and Uncertainty

The next step after computing the predictive posterior, as discussed in 4.1.1, is to compute uncertainty measures. With $y_t = \hat{\boldsymbol{f}}(x|\mathbf{W}_t)$ for each of the T stochastic runs through the network, the predictive mean can be computed via

$$\mathbb{E}[\mathbf{Y}|\mathbf{X}] = \frac{1}{T}\sum_{t=0}^{T} \hat{\boldsymbol{f}}(\mathbf{X}|\mathbf{W_t}).$$

Then, the predictive variance is computed as

$$\mathrm{Var}[\mathbf{Y}|\mathbf{X}] = \frac{1}{T}\sum_{t=0}^{T} f(\mathbf{X}|\mathbf{W_t})^2 - \mathbb{E}[\mathbf{Y}|\mathbf{X}]^2.$$

Simply, T passes through the network are required to estimate the posterior distribution and produce its moments. It is using these moments that we construct uncertainty measures, such as variation ratios, presented in the next section.

4.2 Uncertainty Scoring Functions

In this section, this thesis presents the uncertainty measures used as the scores in the active learning step. The first three employ Bayesian deep learning as formulated in section 4.1. Restated from 2.4.2, in the end, given a model, M, and inputs, $x \in D_{pool}$, the acquisition function in the active learning step will use these scores, $a(x, M)$ to chose which instances to query next, as

$$x^* = \mathrm{argmax}_{x \epsilon D_{pool}}\, a(x, M).$$

4.2.1 Variation Ratios

The notion of variation ratios was discussed previously in 2.4.2 (please see this section for the variation-ratio formula which is used as the acquisition function, $a(x, M)$). The goal with this measure is to select the instance whose classification with the highest probability is the smallest [19]. This ultimately measures lack of confidence. The acquisition function, $a(x, M)$, selects the instances that have the highest variation-ratios scores.

4.2.2 Hybrid

The second uncertainty measure we attempt takes a hybrid approach. We combine both the variation-ratio score with the predictive variance, a metric often used to measure model uncertainty. Inspired by [40], we multiply the former and latter to produce the hybrid score. The acquisition function, $a(x, M)$, selects the instances that have the highest hybrid scores.

4.2.3 Core-Set

Diversity in observations is what motivates the core-set approach. In [33], the authors highlight that query strategies selecting just the most informative scores are susceptible to selecting similar instances, such as consecutive images in a sequence. This leads to an inefficient use of training resources and time. Instead, it's desirable to strive for diversity in the query batch. In [61], the authors formulate this goal as a core-set selection problem: can you select a subset of instances to query such that a model learned over this subset is competitive across all the whole dataset. They achieve this by using the geometry of the datapoints to establish a bound between the average loss over any subset of the dataset and the remaining data. Using this upper bound, the problem becomes one equivalent to the k-Center problem. The k-Center problem can be intuitively understood as seeking the b center nodes for b clusters such that the maximum delta between any datapoint and its respective center node is minimized [61].

Algorithm 2: k-Center-Greedy

Input: data x_i, esisting pool s^0 and a budget b Initialize $s = s^0$

repeat
  $u = \text{argmax}_{i \epsilon [n] \setminus s} \min_{j \epsilon s} \Delta(x_i; x_j)$
  $s = s \cup \{u\}$
until $|s| = b + |s^0|$
return $s \setminus s^0$

In [33], the authors demonstrate use of the greedy implementation of the k-center problem. They select the "centroid" c_i according to:

$$c_i = \arg \max_{x \epsilon X} \min_{c \epsilon C} s(x) * d(x, c)$$

This is the acquisition function, $a(x, M)$. Here, the terms are defined as follows:

- d is the cosine similarity $d_{ij} = \frac{e_i e_j}{||e_i||_2^2 ||e_j||_2^2}$, where e is the image embedding. This embedding is used to formulate the image as a data point in vector space so as to be able to compute its distance to other images. We retrieve this embedding via the final layer of the network's encoder 4.3.

- s represents the uncertainty score of the sample. In this approach, the distance metric is weighted by the uncertainty score. In this thesis, $s = \sum_{pixels} \mathbb{H}(p_c)$, or total entropy, where $\mathbb{H}(p_c) = p_c \log p_c + (1 - p_c) \log(1 - p_c)$, and p_c represents the probability that a pixel takes class c.

Using this greedy implementation, the uncertainty/informativeness score is $s * d$. Instances queried will be those with the highest score.

4.2.4 Evidence

The next metric for uncertainty we experiment with stems from the regime of evidence. Evidence explicitly models weight uncertainties via the theory of subjective logic. Ultimately, Dirichlet distributions are placed on class probabilities, and the network's predictions are considered subjective opinions. The network is used to learn the function that amalgamates evidence that generates these opinions [62]. Evidence

is valuable for uncertainty estimation because, while like using Dropout, evidential methods can be fast, unlike Dropout, evidential methods are calibrated.

Overall, evidential learning addresses a weakness in softmax probabilities. Softmax tends to inflate the probability of the predicted class, due to the exponent term used in the outputs of neural networks. The Dempster-Shafer Theory of Evidence (DST) generalizes Bayesian theory and assigns belief masses to subsets of frames of the powerset of propositions (e.g. a class label). These belief masses are computed as a function of the evidence supporting a specific proposition and nothing more. A formalization called Subjective Logic (SL) elucidates the belief assignment over "frames of discernment" as a Dirichlet Distribution. A belief mass assignment, or a "subjective opinion", can be formulated as a Dirichlet distribution with parameters $\alpha_k = e_k + 1$ [62], where e is the evidence. The parameter set of a categorical distribution, which softmax serves to output, is replaced by the parameter set of a Dirichlet distribution. Thus, the output itself serves to parameterize a distribution over possible softmax outputs as opposed to a point estimate of said softmax outputs. To achieve this task one must simply modify the loss function and optimize the model using standard backpropegation.

In the end, uncertainty is computed as $u = K/S$. $S = \sum_{i=1}^{K} \alpha_i$ is called Dirichlet strength. K, is the number of "singletons" in the "frame of discernment" (in our case, this is merely 2, as we are only dealing with class cloud and non-cloud). Thus, for each pixel, instead of a softmax density, we have the parameters for a Dirichlet which we can plug into the equation for u, and sum u (uncertainty) across all pixels to generate the uncertainty for a given image. The acquisition function, $a(x, M)$, ultimately selects the instances that have the highest uncertainty scores, u.

4.2.5 Random

The final uncertainty measure is a baseline based upon which the previously discussed measures will be compared. This metric places an equal score on every image instance. Using this score in the acquisition function is akin to placing a uniform distribution over all possible points in the random pool [24].

4.3 ENet

While the acquisition function is crucial for an active learning system implementation, at the end of the day, the model is the subject of the learning process. It was thus important to tactfully consider the model architecture we would use. Given that this task is, at it's core, a segmentation task, it made sense to consider the state-of-the art segmentation network architectures. As discussed in 2.6, some of the most popular include DeepLabv3+ [11], Mask R-CNN [34], and U-Net [59]. We decided it best to experiment first with an architecture that was simpler. A simpler architecture is easier to train and would allow us to iterate faster. Furthermore, even though the question at hand is about minimizing the number of annotator clicks, this focus stems from a desire to decrease the time to generate the accurate and robust dataset. A simpler architecture would also be faster at inference time. U-Net is the simplest of the three, and thus we decided to turn here first. We ultimately settled on ENet, a derivative of the U-Net architecture family.

4.3.1 Motivation

ENet (literally, Efficient Neural Network) is exactly as its name implies. ENet's authors designed the network to address the problem that many segmentation networks have been sustaining: prolonged run times due to mass floating point operations. ENet is specifically designed for tasks which demand low latency. In our case, this permits faster learning and inference when the model is ultimately used to help generate the cloud mask dataset. ENet is 18 times faster, requires 75 times less floating point operations and has 79 times less parameters while achieving similar or better accuracy than SegNet, another popular U-Net derivative [57].

4.3.2 The Network

The ENet network, detailed in 4-2, consists of 28 blocks which comprise an encoder and then decoder framework. It adopts a notion from ResNet by employing a main branch in addition to extension branches which are eventually merged via element-

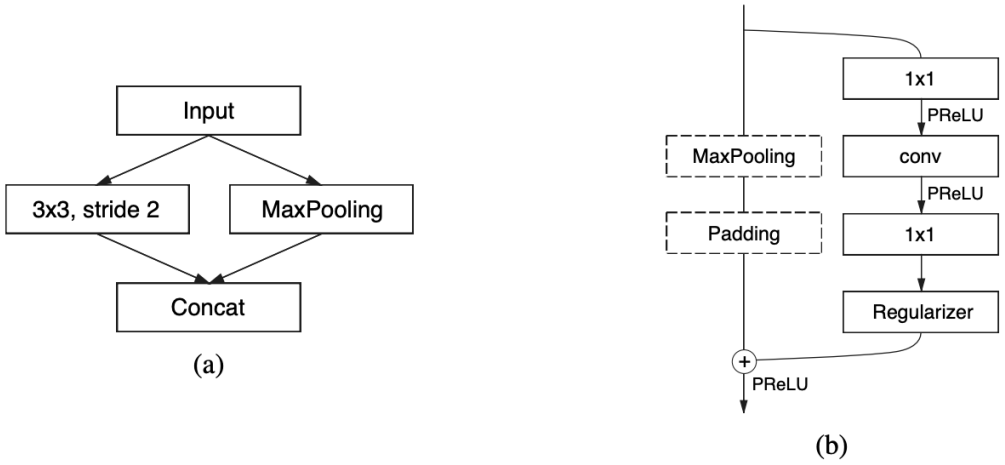


Figure 4-1: (a) ENet initial block. Max Pooling is performed with non-overlapping 2×2 windows, and the convolution has 13 filters, which sums up to 16 feature maps after concatenation. This is heavily inspired by [66]. (b) ENet bottleneck module. Convolution is either a regular, dilated, or full convolution (also known as deconvolution) with 3×3 filters, or a 5×5 convolution decomposed into two asymmetric ones [57].

wise addition. Each block (known as a "bottleneck block", see 4-1) is made up of 3 convolutional layers. The first layer consists of a 1x1 projection, used for dimensionality reduction. The second is the main convolutional layer. Finally, there is a 1x1 expansion layer. ENet uses batch normalization between all convolutions and uses spatial dropout after each block for regularization. For the purposes of using dropout to achieve a Bayesian neural network, as discussed previously in the chapter, these dropout layers remain "on" during inference time. Consistent with the ENet paper, the dropout probability, p, is set to 0.01 before the bottleneck 2.0, and $p = 0.1$ after. Furthermore, as in the paper, spatial dropout is used, as opposed to element-wise dropout. Normal, element-wise dropout can suffer in computer vision applications as early network layers may encourage learning of feature maps with strong correlation. Spatial dropout on the other hand can achieve the same goals as normal dropout while encouraging independence in between said feature maps [3]. 4-3 illustrates spatial dropout.

Name	Type	Output size
initial		$16 \times 256 \times 256$
bottleneck1.0	downsampling	$64 \times 128 \times 128$
4× bottleneck1.x		$64 \times 128 \times 128$
bottleneck2.0	downsampling	$128 \times 64 \times 64$
bottleneck2.1		$128 \times 64 \times 64$
bottleneck2.2	dilated 2	$128 \times 64 \times 64$
bottleneck2.3	asymmetric 5	$128 \times 64 \times 64$
bottleneck2.4	dilated 4	$128 \times 64 \times 64$
bottleneck2.5		$128 \times 64 \times 64$
bottleneck2.6	dilated 8	$128 \times 64 \times 64$
bottleneck2.7	asymmetric 5	$128 \times 64 \times 64$
bottleneck2.8	dilated 16	$128 \times 64 \times 64$
Repeat section 2, without bottleneck2.0		
bottleneck4.0	upsampling	$64 \times 128 \times 128$
bottleneck4.1		$64 \times 128 \times 128$
bottleneck4.2		$64 \times 128 \times 128$
bottleneck5.0	upsampling	$16 \times 256 \times 256$
bottleneck5.1		$16 \times 256 \times 256$
fullconv		$C \times 512 \times 512$

Figure 4-2: ENet architecture. Output sizes are given for an example input of 512×512. [57]

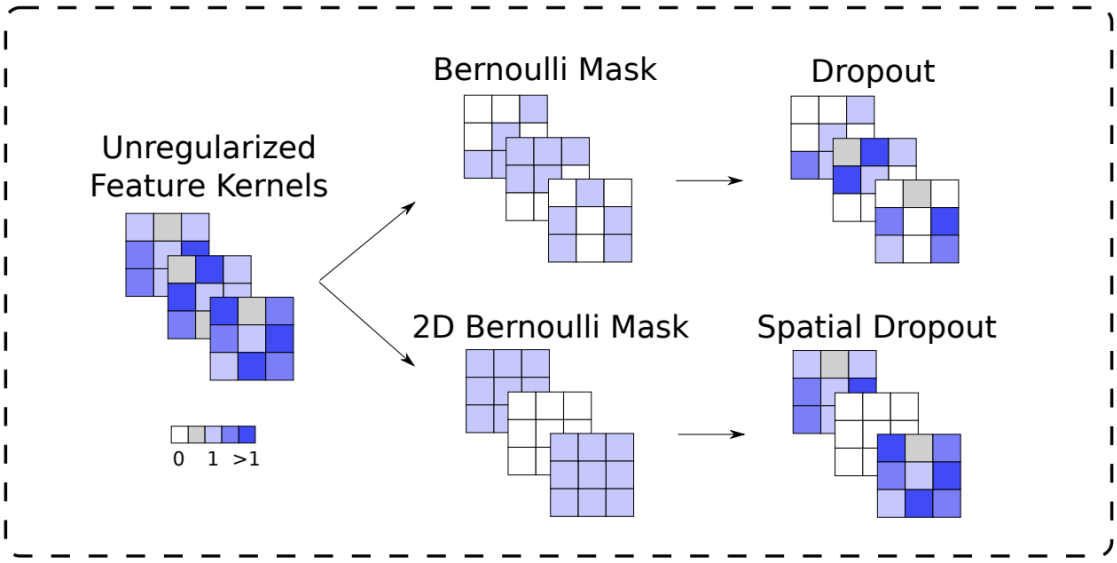


Figure 4-3: Element wise dropout vs 2D spatial dropout [3].

4.4 Algorithm Overview

The query selection step of the active learning system begins when a subset of the entire dataset, data which are accompanied by annotations, d_{init}, are fed to the ENet model, as d_{train}, for training. Subsequently, the system runs the trained model on the rest of the unlabeled data pool, d_{pool}, providing predictions for each instance. Via the various scoring functions discussed in 4.2, each instance is given an informativeness (uncertainty) score. We choose the instances with the top 10% highest scores, b, to be queried to the oracle for labeling. b is equaled to $0.1 * ||d_{pool}||$. Once the queried instances are annotated, these instances are added to d_{train}, and the loop begins again. This loop continues until d_{pool} is empty, or until the mask outputs from the model are satisfactory to the annotator.

Algorithm 3: The Active Learning Query Selection Step

Input: A dataset, D, of size n, where a subset of D, d_{init}, approximately of size $0.01 * n$, is accompanied with annotations, while the rest of the data, d_{pool}, have no labels.

Output: A trained ENet model, M

Initialize the ENet model, M

$d_{train} \Leftarrow d_{init}$

Train M using d_{train}

while *size(d_{pool})$\neq$ 0 AND annotator desires to continue training the model* **do**

- $\mathsf{y}^* \leftarrow M(d_{pool})$; prediction
- $\mathsf{u}^* = \emptyset$
- **for** *i = 1 to b* **do**
  - $\mathsf{u}^* \leftarrow \mathsf{u}^* \cup argmax_{(x \in d_{pool}, y^*)}(\phi(\mathsf{y}^*))$; uncertainty measurement/scoring
- `/* Scoring function, φ, chosen from among: variation-ratios, hybrid, core-set, evidence, and random approaches. b is the number of instances to query each round. */`
- $\mathsf{q}^* \Leftarrow$ annotator(u^*)
- $d_{pool} \Leftarrow d_{pool} \setminus \mathsf{u}^*$
- $d_{train} \Leftarrow d_{train} \cup \mathsf{q}^*$
- Train M using d_{train}

4.5 Summary

This chapter presents the use of uncertainty sampling as the driver for the query step in our system's active learning loop. Recall, the active learning step in this system is used to learn from the accurate image-mask pairs generated by the expert annotator. We use active learning because previous works have shown that it can grant the ability to achieve greater accuracy learning off of fewer samples if the network is able to choose the instances it learns from. This amounts to less annotation work for the annotator (ideally, a significant order of magnitude less). We turn to uncertainty, or a model's predictive confidence, as the informativeness measure when selecting instances to query to the oracle/annotator. This decision is motivated by previous work indicating that neural networks' prime weaknesses are out-of-distribution instances. We interest ourselves in measures of uncertainty like variation-ratios, a hybrid of variation-ratios/predictive variance, a core-set metric, and evidence. We expect to see an improvement in accuracy using these metrics as opposed to a random query strategy or merely training without active learning altogether.

Chapter 5

Experimental Setup

We present 3 experiments in this thesis: pixel classifier evaluation, evaluation of the best pixel classifier with automatic detectors (both detailed in chapter 3), and evaluating the scoring functions (detailed in chapter 4).

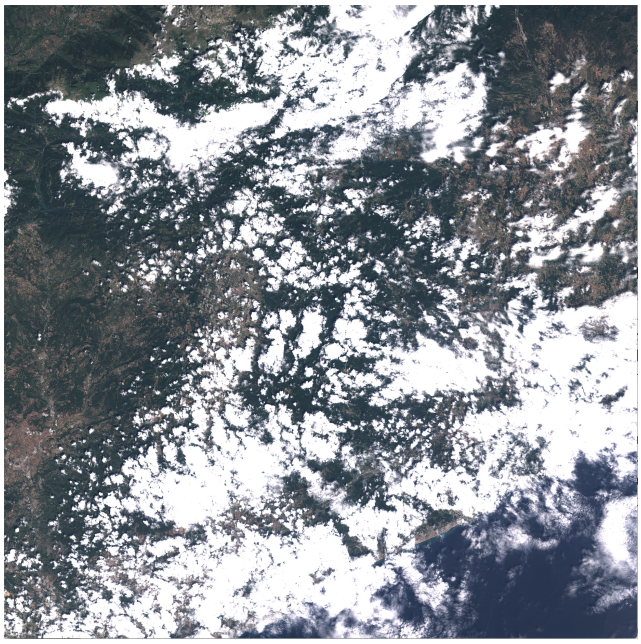
5.1 The Datasets

5.1.1 Pixel Classifier Evaluation

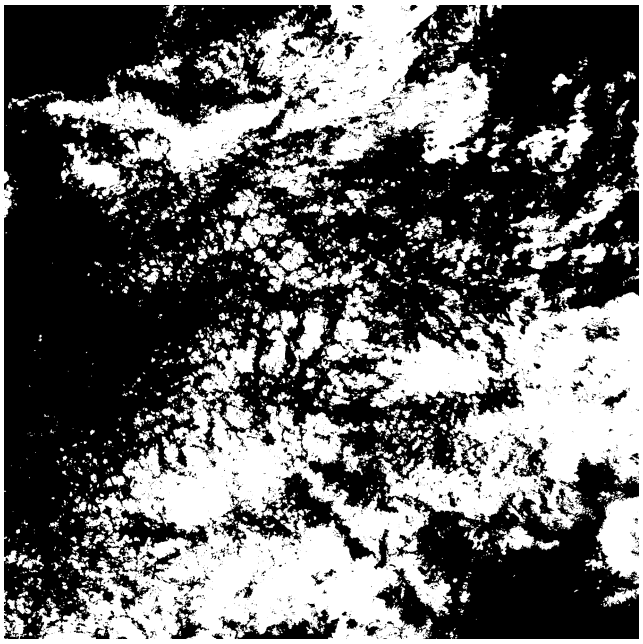
In this experiment, we used four image scenes with 5, 1, 4, and 2 time-series images respectively. Our dataset was comprised of 12 images and associated masks. Only "cloud" and "background" annotations were made. A sample image scene is seen in Figure 5-1.

5.1.2 Comparison with Automatic Detectors

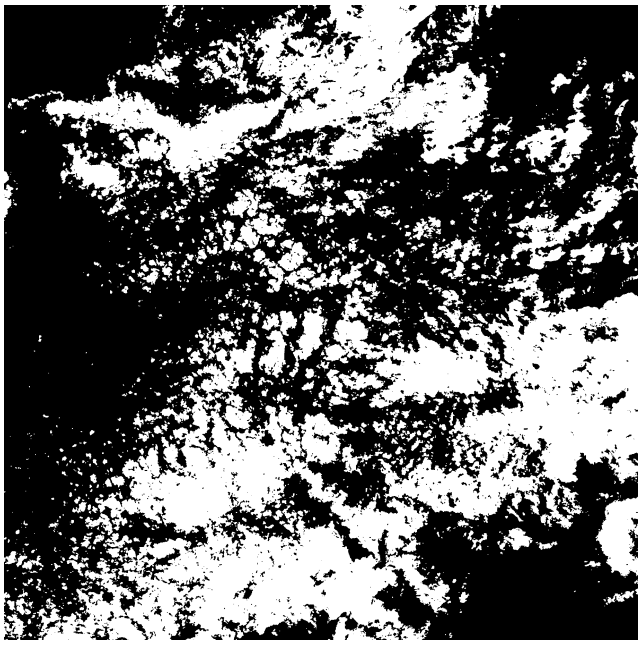
In this second experiment, we used a dataset of 12 masks and associated images: 12 separate scenes captured on arbitrary dates. Again, only "cloud" and "background" annotations were made. A sample image scene is seen in Figure 5-1.



(a) Original image.



(b) Cloud mask ground truth using BMM.



(c) Cloud mask ground truth using RF.

Figure 5-1: An example image scene (5-1a) and corresponding masks (generated by BMM in 5-1a and RF in 5-1a) in the datasets from the pixel classifier evaluation and comparison with automatic detector experiments.

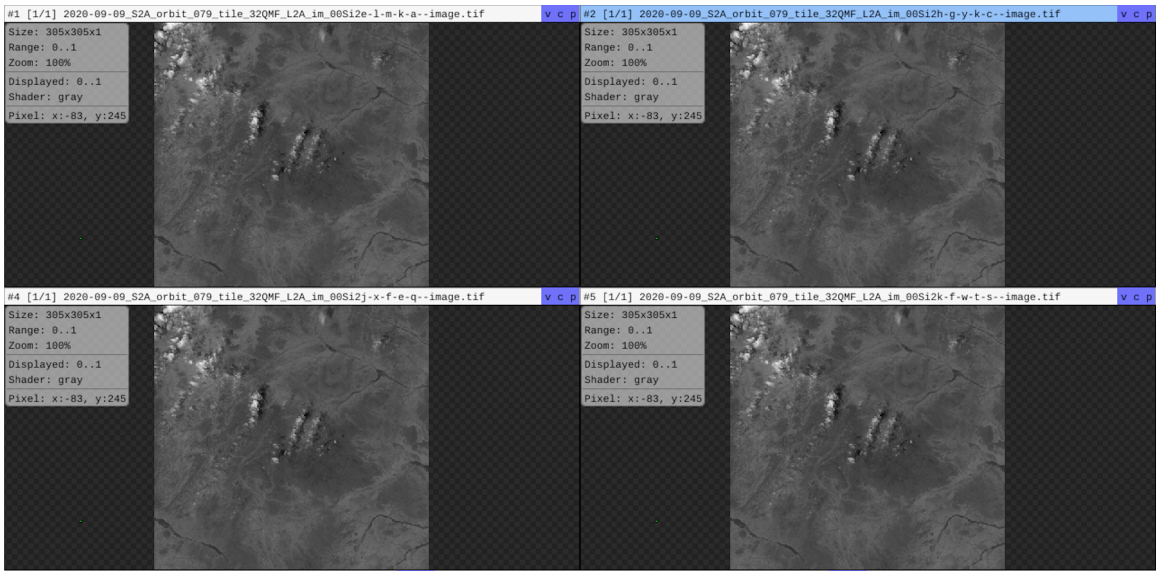


Figure 5-2: A single image scene from the dataset used in the scoring function evaluation experiment. Displayed is a subsample of 4 out of the 14 perturbations made for the image. In this subsample, the perturbations only consist of contrast changes.

5.1.3 Scoring Function Evaluation

We ran this experiment on 100 Sentinel-2 image scenes of size 610 x 610; each image consisted of 12 channels (the Sentinel-2 8A band was not available in the dataset, however we have no reason to believe the absence of the band changes the trends of the results). Prior to running the active learning algorithm, we conducted preprocessing. We started with the 100 scenes (referred to as superimages) and extracted 4 non-overlapping subimage blocks of size 305 x 305. This brought the dataset to 400 images. Each image was augmented by 4 rotations per horizontal flip orientation (thus 8 rotation/flip augmentations total); each of these augmented images were augmented further by 3 contrast changes. Of these 24 perturbations, 14 were selected at random to be included in the dataset. This resulted in a 5,600 image (subimage) dataset. Figure 5-2 displays a subsample of 4 out of the 14 perturbations made for a single image where the only perturbation is a contrast change.

5.2 Objectives and Evaluation Procedure

5.2.1 Pixel Classifier Evaluation

In the first experiment, we compared the different methodologies: SVMA, RFA, and BMMA with pixel inputs, aiming to minimize the number of clicks needed to reach a 90% and then 95% accuracy. More concretely, if the ground truth error percentage between the two applications of the same process (BMMA or RFA) was 5%, this meant that it had 95% precision. Thus, the target accuracy for the experiment using a given methodology to achieve 90% accuracy relative to this ground truth (and its respective error) meant getting $0.9 * 95\% = 85.5\%$ accuracy compared to the ground truth. The experiment thus evaluated how many clicks were required to achieve 85.5% accuracy compared to the ground truth. In all experiments, we measured experimental results against the first trial of the ground truth. This thesis originally strove to annotate to achieve 99% accuracy instead of only 95% and 90%. However, when running the experiment, it became apparent that without further optimizations and modifications, it was not possible to get that high accuracy. Instead of continuing with several image instances where a given methodology could not achieve the 99% accuracy, we decided to merely stop at 95%.

We used each methodology to annotate the image and generate the best possible cloud mask. We then compared the output mask to the ground truth. The annotator stopped the annotation process upon reaching the 90% and then 95% goal accuracy relative to the ground truth. Finally, we recorded the total number of clicks made.

We trained the BBMA process with pixel inputs such that each pixel only consisted of one band: B1 or B11. B1 is the "coastal aerosol" band with wavelength of approximately 0.443 µm. B11 is the SWIR band with wavelength of approximately 1.610 µm [39]. Empirically, we found that using solely these two bands resulted in the best cloud masks. For the BMMA processes, we used a Dirichlet distribution prior for the weights distribution.

5.2.2 Comparison with Automatic Detectors

Next, we compared BMMA using pixel inputs and BMMA using patch inputs to existing automatic cloud detectors. As before, the BMMA process using pixel inputs used only B1 or only B11. The BMMA process with patch inputs took the RGB channels (B4, B3, and B2) as input. We compared against a parallax cloud detector PCD, [13], and two cloud detectors based on the analysis of time series, GTS [28], and DTS [14]. For this comparison, we expanded the dataset to include twelve image scenes. For each scene, there was a single image of interest and ground truths generated by 3.2.1 and 3.2.2. We completed BMMA by applying a morphological area opening. The GTS and DTS automatic detectors took in a time-series from the given scene and generated the cloud mask for the image of interest. The PCD method instead ran solely on the image of interest. The Sentinel-2's Sen2cor software [51] also provides cloud masks of its own. We compared the cloud masks of these four methods to both ground truths through the percentage of error score.

5.2.3 Scoring Function Evaluation

In this experiment, we explored the impact active learning has on achieving the goal of decreasing annotation time and we identify the optimal scoring function to achieve this. Plotting the learning curve - the evolution of the intersection over union (IoU) - of each scoring function, we compared each of the scoring functions from 4.2 and used the random sampling strategy, 4.2.5, as the standard upon which to compare.

The Active Learning Setup

The active learning experiment proceeded by taking 10% chunks of the dataset to query/learn from after each iteration. Concretely, if the total number of images in the data pool was n, the model began by learning on a random chunk that was size $0.1 * n$. At each step, the active learning system selected the top $0.1 * n$ most uncertain samples from the unlabeled data pool to get labeled by the oracle. The system then added this data to the existing labeled data that the model would learn

in the next iteration. There were thus 10 active learning steps (including the original learning step on a random batch of data). In the first part of the experiment, the entire dataset consisted of images and ground truth masks provided by the Sentinel-2 Level-2A Algorithm. Thus, when the active learning step chose images to query, the stored masks were merely retrieved from the data store (serving as the oracle) and these image-mask pairs were used for training. In the second part of the experiment, while we still used the same image scenes, the masks for the training data came from the fast supervised cloud mask generation process in 3.

Acquisition Functions

The Variation-Ratios, Hybrid, and Random approaches are relatively straightforward and need no further specification. In 4.2.3, we highlight that the greedy implementation of the k-center problem (k-Center-Greedy) was used for the Core-Set approach. This algorithm is detailed in 4.2.3. Additionally, to improve the speed of the D matrix computation needed to determine the cosine similarity between every image embedding vector, we cut down the dimensionality of the embeddings by averaging across the feature maps, and we vectorized the cosine similarity computation.

For the Evidence approach, implementation consisted of changing the output layer to the ENet network and modifying the loss function. We set the output activation for ENet to Softplus to ensure non-negative output. Again, this output served as the evidence vector for a Dirichlet distribution. The loss function is implemented as

$$\mathcal{L}_i(\Theta) = \sum_{j=1}^{K}(y_{ij} - \hat{p}_{ij})^2 + \frac{\hat{p}_{ij}(1-\hat{p}_{ij})}{(S_i+1)},$$

$$KL[D(\boldsymbol{p_i}|\tilde{\boldsymbol{\alpha}}_{\boldsymbol{i}})||D(\boldsymbol{p_i}|\mathbf{1})] = \log$$
$$(\frac{\Gamma(\sum_{k=1}^{K}\tilde{\alpha}_{ik})}{\Gamma(K)\prod_{k=1}^{K}\Gamma(\tilde{\alpha}_{ik})}) + \sum_{k=1}^{K}(\tilde{\alpha}_{ik}-1)[\psi(\tilde{\alpha}_{ik}) - \psi(\sum_{j=1}^{K}\tilde{\alpha}_{ij})],$$

$$\mathcal{L}(\Theta) = \sum_{i=1}^{N}\mathcal{L}_i(\Theta) + \lambda_t \sum_{i=1}^{N} KL[D(\boldsymbol{p_i}|\tilde{\boldsymbol{\alpha}}_{\boldsymbol{i}})||D(\boldsymbol{p_i}|\langle 1,...,1\rangle)].$$

$S = \Sigma_{i=1}^{K}(e_i + 1)$, representing Dirichlet strength. Also recall α_k are the Dirichlet parameters.

ENet

A few intricacies were necessary in the ENet architecture for the experiment. First, for the Variation-Ratios, Hybrid, and Core-Set approaches, dropout remained on during inference time. This remained off, however, for the Evidence and Random approaches. For the Core-Set approach, we added a hook to extract intermediate features (the output features of the encoder) for the image embedding in the Core-Set function. The system optimized the model using the Adam optimizer. We used the same custom weighting scheme as is used in the ENet paper, $w_{class} = \frac{1}{ln(c+p_{class})}$ [57].

Chapter 6

Results

In this chapter, we presents the results of our extensive experimentation. Ultimately, this chapter sheds light on the classifier method best suited for the fast supervised cloud annotation task and the uncertainty scoring function that elicits the best results from the active learning step.

6.1 Pixel Classifiers

The experiment comparing pixel classifiers measured the average number of clicks/annotations an annotator needed to achieve 90% accuracy and then 95% accuracy compared to the ground truth. This experiment showed that the BMMA process (using pixel inputs) performs better than SVMA and RFA. **Table 6.1** displays our two recorded statistics. We computed the average number of clicks made across the dataset for each methodology. On average, it took less than 1 click to achieve the desired percentage of ground truth accuracy for BMMA, whereas it took over 2 on average for RFA and greater than 3 for SVMA. Thus, for the fast supervised cloud annotation step, the best classifier algorithm option is indeed a clustering based model, specifically a BMM. Compared to using other classification algorithms, using the BMM, one can learn from minimal clicks and generate rather accurate cloud masks.

	90% g.t. accuracy		95% g.t. accuracy	
	BMMA	RF	BMMA	RF
SVM	3.8 ± 1.8	3.8 ± 1.9	4.7 ± 2.9	4.3 ± 2.8
RF	2.3 ± 0.5	2.2 ± 0.4	2.4 ± 0.7	2.3 ± 0.5
BMMA B1	1.3 ± 0.6	1.2 ± 0.6	1.4 ± 0.8	1.3 ± 0.7
BMMA B11	1.0 ± 0.1	1.2 ± 0.6	1.1 ± 0.3	1.3 ± 0.7

Table 6.1: Average number of clicks (with standard deviation) to achieve goal percentage ground truth (g.t.) accuracy, relative to RF and BMM generated ground truth. Clicks are averaged across each methodology (SVMA, RFA, BMMA using only B1, and BMMA using only B11).

6.2 Automatic Detectors

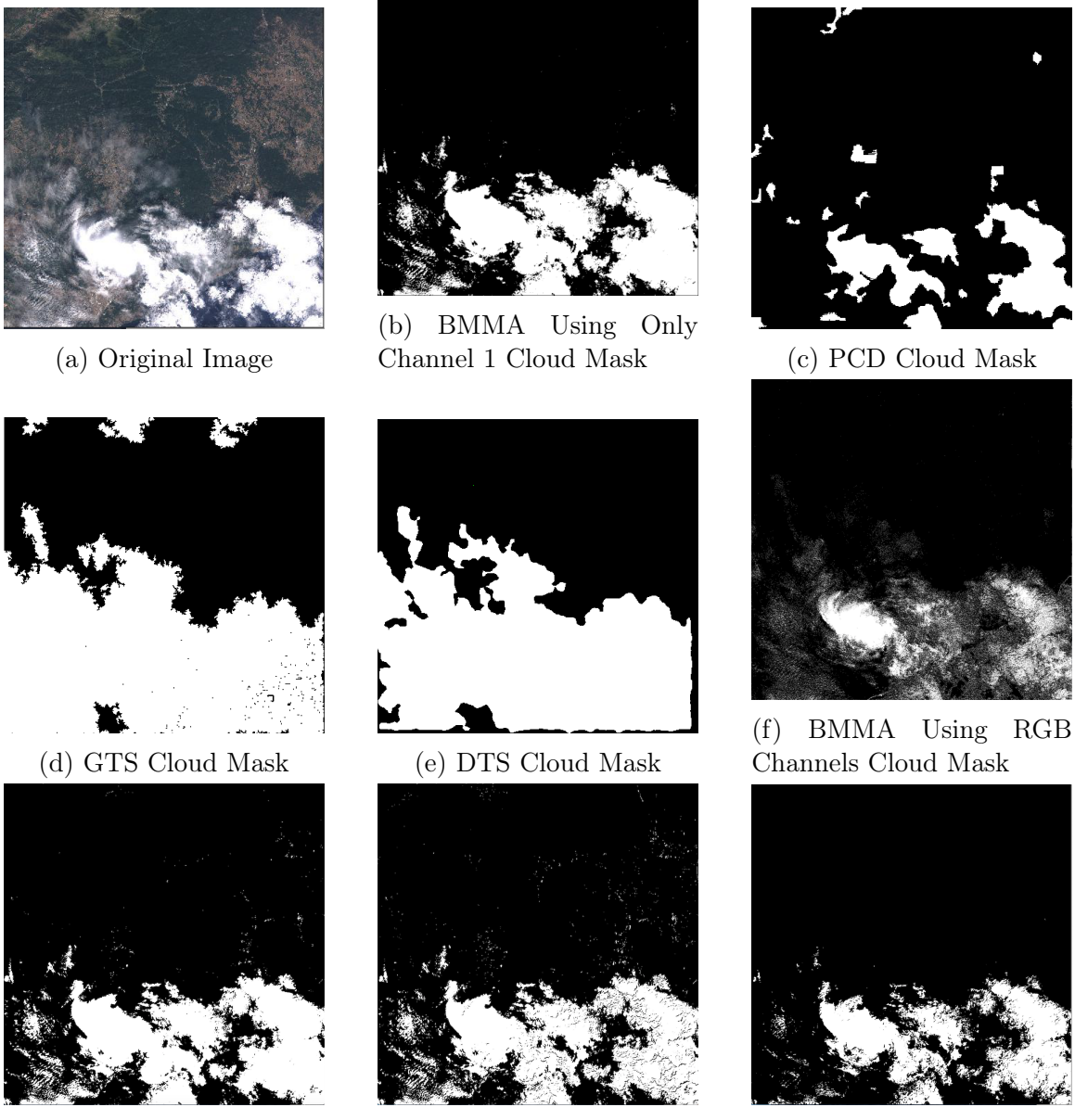
The experiment comparing with automatic detectors measured the accuracy of each mask generation method compared to ground truth. This accuracy is reflected as an error percentage. This experiment shows that BMMA with pixel inputs performs better than BMMA with patches, the existing automatic detectors, and the cloud masks provided by Sentinel-2. **Table 6.2** records the percentage error each cloud mask generating process achieves with respect to the RF and BMM ground truths. Unlike in experiment 1, the goal was not to get to a particular percentage accuracy compared to ground truth, but was to get the smallest error possible. BMMA using B1 led to the smallest error when compared to ground truth. Figure 6-1 shows sample outputs from each of the methodologies. Thus, the BMMA process is a supervised process that empowers an expert annotator to generate cloud masks superior to those of powerful automatic detectors.

6.3 Adding Open Morphology

This experiment shows both the impact of the open morphology used on the best method (BMMA using B1) and the average number of clicks needed to achieve this superior performance. **Table 6.3** records average error of BMMA using B1 with respect to the DP and RF ground truths respectively. It also demonstrates an 1%

	Error w.r.t RF Ground Truth	Error w.r.t BMM Ground Truth
BMMA using channel 1	**5.19%**	**1.56%**
BMMA using channel 11	16.32%	16.08%
grompone-timeseries-v2	11.10%	13.33%
dagobert-timeseries-v2	11.09%	13.32%
parallax-cloud-detector	7.81%	9.90%
Sentinel-2 Cloud Mask	5.20%	7.70%
BMMA using RBG channels	8.61%	9.34%
BMMA using PCR and patch size 3	8.31%	7.25%
BMMA using PCR and patch size 5	20.23%	20.03%
BMMA using PCR and patch size 7	22.93%	22.54%
Ground truth self-comparison	3.30%	1.80%

Table 6.2: Percentage error each cloud mask generating process achieves with respect to the RF and BMM ground truths. Compares the BMMA processes using only channel 1, only channel 11, only RGB channels, and only RGB channels using patches and PCR, to automatic detectors.



(a) Original Image

(b) BMMA Using Only Channel 1 Cloud Mask

(c) PCD Cloud Mask

(d) GTS Cloud Mask

(e) DTS Cloud Mask

(f) BMMA Using RGB Channels Cloud Mask

(g) BMMA Using PCR and Patch Size 3 Cloud Mask

(h) BMMA Using PCR and Patch Size 5 Cloud Mask

(i) BMMA Using PCR and Patch Size 7 Cloud Mask

Figure 6-1: A comparison of each of the cloud mask generators (generated by our annotation system, b, and the other automatic detectors, c-e).

	Error	Averages
BMM GT	without opening	2.83% ± 1.58%
	with opening	1.32% ± 1.61%
RF GT	without opening	4.93% ± 4.93%
	with opening	4.00% ± 4.00%
	Number of clicks	2.42 ± 1.00
	Number of clusters	6.79 ± 0.80

Table 6.3: Average error (with standard deviation) of BMMA using B1 with respect to the BMM and RF ground truths respectively. Averages are reported after applying a morphological opening or without it. The average number of clicks to achieve the reported error and the average number of clusters in the mixture model are also recorded.

improvement when a morphological opening is used. We also recorded the average number of annotations needed to achieve the reported error and the average number of clusters of the mixture model. As the error reflects, we were able to get up to around 98% accuracy with respect to the BMM ground truth and 96% with respect to the RF ground truth. On average, the annotator made 2.42 annotations. This means that to improve from the 95% accuracy reported in **table 6.1**, only about 1-2 more clicks are needed!

6.4 Query Frameworks

In this experiment that compares the uncertainty scoring functions, we reported model loss and intersection over union (IoU) learning curves. Specifically, we plot the model loss evaluated on the test data set versus the percentage of training data that was used to train said model. The loss is evaluated and recorded after each active learning step, during which the next 10% of additional training data is selected and added to the training data batch. Said data to be added to this training batch is selected using one of the variation ratios, hybrid, core-set, evidence, or random uncertainty scoring functions. Figure 6-2 compares the loss learning curve of these functions. With respect to IoU, in Figures 6-3, 6-4, and 6-5 we report, respectively, the "non-cloud" class, "cloud" class, and mean IoU evaluated on the test data set versus the percentage

of training data used to train said model. Each IoU is evaluated and recorded after each active learning step. This experiment shows that the hybrid uncertainty scoring function performs best compared to the others in our active learning task.

We first look at the reported model loss vs percentage of train data used. All functions, except evidence, begin above 0.6 and, at some point, all converge below 0.5. The hybrid and core-set seem to converge to the lowest error, just below 0.4, again not including the evidence function. As discussed in Chapter 4, the evidence function comes with a unique loss function. This function differs from the cross-entropy loss used for each of the other functions. The evidence function's loss starts at 0.346. and decreases only down to 0.336. We find more interesting results when evaluating IoU.

In Figure 6-3, we see that the hybrid "non-cloud" class IoU learning curve peaks at 50% of training data used. This is the most prominent peak of all the curves, not including evidence, measuring approximately 0.92. This means that in the context of classifying the "non-cloud" class, when using the hybrid uncertainty function, we select new instances that are informative enough such that model performance peaks after only seeing 50% of the training dataset. In this same graph, the curve for evidence appears to be more promising, as it begins just above 0.91 and increases to above 0.94. However, in the context of this experiment, the strength of this result takes a different light when considering the "cloud" class IoU.

In Figure 6-4, we see that the hybrid "cloud" class IoU learning curve peaks at 50% of training data used. This is the most prominent peak of this curve, and measures approximately 0.31. This means that in the context of classifying the "cloud" class, when using the hybrid uncertainty function, we select new instances that are informative enough such that model performance peaks after only seeing 50% of the training dataset. For the hybrid "cloud" class IoU learning curve, one sees clearly that the hybrid function outperforms all others. One sees as well that the evidence curve straggles further below the others. This informs us that, as currently designed, the evidence approach tends to favor labeling pixels as "non-cloud" as opposed to "cloud" when comparing against the other approaches.

Looking at the mean IoU, in Figure 6-5 we again see a prominent peak for the hybrid function at 50% of training data used; this peak measures in at approximately 0.62. These data tell us that the hybrid approach performs the best, and permits us to achieve an IoU superior to all other scoring functions (even random) using the least amount of training data. The fact that hybrid achieves this superior score prior to the model training on the entire dataset shows that using active learning, we can achieve better model accuracy learning on fewer instances. The fact that the hybrid approach performs better than the random approach demonstrates that cleverly selecting a scoring function is superior, in this active learning regime, to selecting arbitrarily. Ultimately, these data demonstrate that by using the hybrid scoring function as our uncertainty measurement, if an annotator employs the active learning system, they can more easily generate a cloud mask dataset. In the case of this experiment, instead of labeling an entire dataset today and labeling a new batch of data in the future, they can label merely 50% of the current data, train an ENet model and have the model label future data and achieve 0.62 mean IoU. We believe that this number will increase significantly if we replace ENet with a superior segmentation network and will explore this in future work.

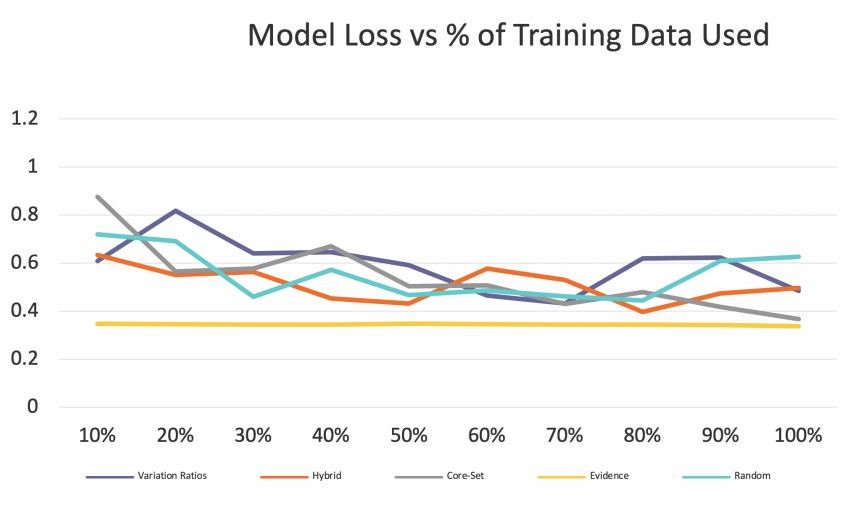


Figure 6-2: Model loss evaluated on the test data set vs the percentage of training data used to train said model. Loss is evaluated and recorded after each active learning step, during which the next 10% of additional training data is selected and added to the training data batch. Said data to be added is cleverly selected using one of the variation ratios, hybrid, core-set, evidence, or random uncertainty scoring functions. The plot compares the loss learning curve of these functions.

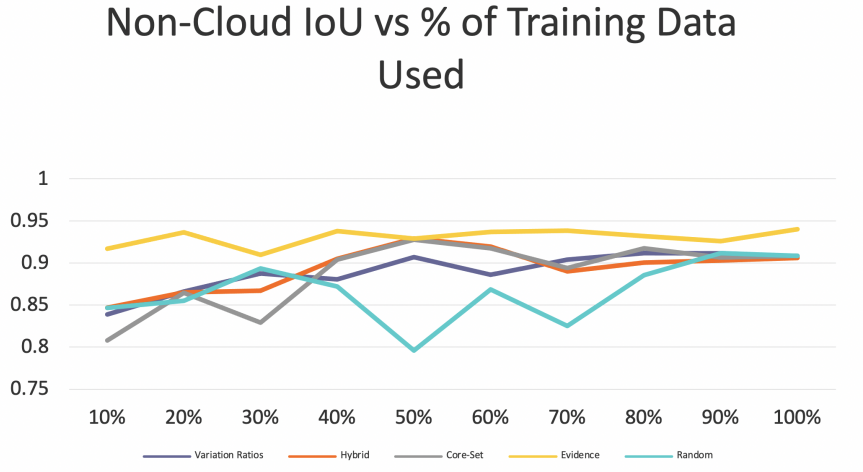


Figure 6-3: "Non-Cloud" class IoU evaluated on the test data set vs the percentage of training data used to train said model. IoU is evaluated and recorded after each active learning step, during which the next 10% of additional training data is selected and added to the training data batch. The plot compares the "non-cloud" class IoU learning curve as a function of the variation ratios, hybrid, core-set, evidence, and random uncertainty scoring functions.

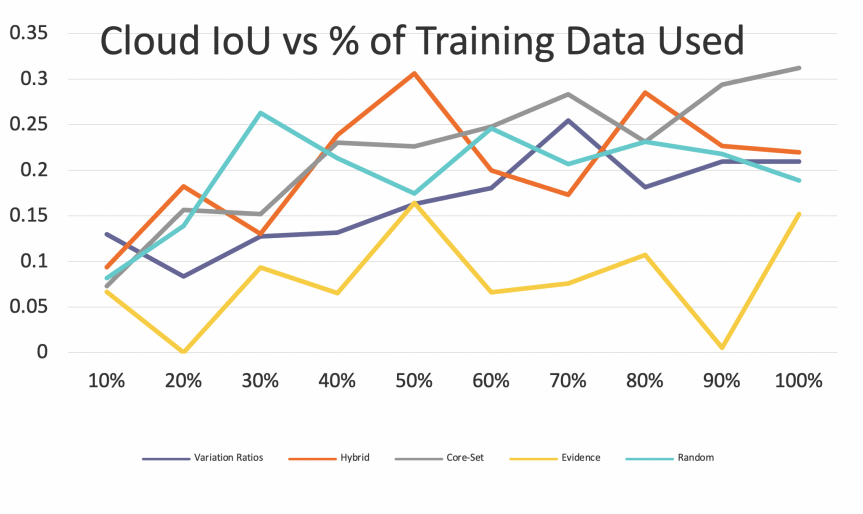


Figure 6-4: "Cloud" class IoU evaluated on the test data set vs the percentage of training data used to train said model. IoU is evaluated and recorded after each active learning step, during which the next 10% of additional training data is selected and added to the training data batch. The plot compares the "cloud" class IoU learning curve as a function of the variation ratios, hybrid, core-set, evidence, and random uncertainty scoring functions.

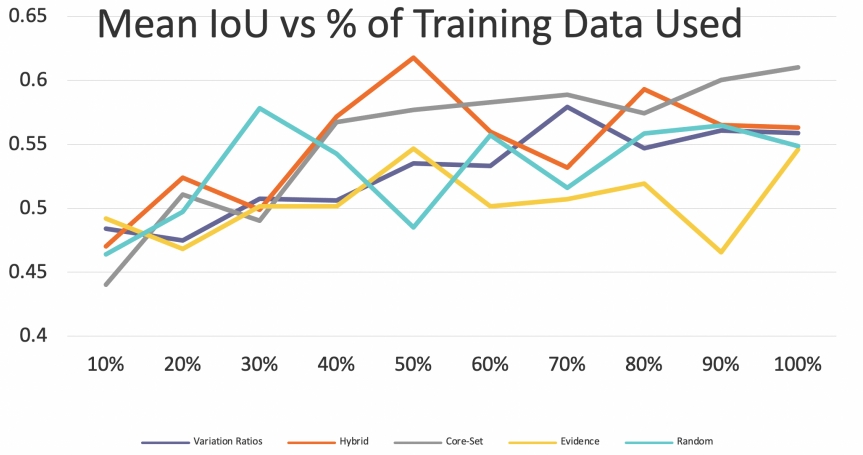


Figure 6-5: Mean class IoU evaluated on the test data set vs the percentage of training data used to train said model. IoU is evaluated and recorded after each active learning step, during which the next 10% of additional training data is selected and added to the training data batch. The plot compares the mean class IoU learning curve as a function of the variation ratios, hybrid, core-set, evidence, and random uncertainty scoring functions.

Chapter 7

Conclusion

This thesis presents a fast supervised cloud mask generation process for Sentinel-2 images that allows expert annotators to quickly generate accurate cloud masks. Many previous pursuits focusing on the task of cloud mask generation go the route of building automatic detectors. These algorithms, however, are not ubiquitously successful, and thus aren't performant enough in many tasks. Our results indicate that by employing relatively minor human intervention (annotators) and intelligent machine learning that force multiplies the annotator's work, one can efficiently generate highly accurate cloud masks. This thesis attempts to generate cloud masks using a system that learns from only a few annotations from an expert annotator, following the application of a BMM. Using the Sentinel-2 LC1 product, we found that using the band B1 as the sole feature works best. This annotation process has been proven to be efficient on Sentinel-2 and its main advantage is that it uses pixel based non-contextual classification. This is thanks to the rich spectral content of each pixel, which seems sufficient to almost always discriminate clouds from ground.

Additionally, this thesis presents a second level to the cloud mask generation system comprised of an active learning feedback loop. To power the active learning loop, we contribute a modified ENet architecture that integrates uncertainty estimation. This information provides a measure of uncertainty with the output and is ultimately what informs the active learning query step. Intelligently querying uncertain instances to be labeled by the annotator, our system learns to generate cloud masks

for an unlabeled image data pool after training on fewer instances than a non-active learning regime.

7.1 Lessons Learned

Working on this thesis, I learned how important it is to consider model uncertainty in any task and use this information to improve decision making. Sometimes you can actually become more "in the know", by understanding what you don't know. This work also taught me the importance of starting problem solving with simpler solutions and iterating from there. Starting with the most complex solution from the onset can lead to unnecessary confusion, complication, and work. Starting simpler and iterating allows for initial, elementary discoveries that help focus attention on where more complex solutions should be applied. Finally, after this thesis, I have greater appreciation for the planning that must go into planning a deep learning experiment. Deep learning experiments are often computationally extensive and require keen use of GPU resources. Running an experiment for the active learning process, however, involved not only considering the model training, but also the uncertainty score computations for the entire unlabeled data pool. This meant considering a whole set of additional computations and data transfers. I realized that GPU resources weren't being efficiently utilized, causing the experiment to run for a very long time. I ultimately had to conduct additional preprocessing prior to running the experiment, had to re-write functions to use GPU computing power, and had to consider efficient CPU usage as well.

7.2 Future Work

In the first step of our future work, we consider further optimizing and experimenting with the evidence uncertainty scoring function approach. In works such as [65], we see the promise of evidential methods for uncertainty measurement and deep learning. We believe with further parameter tuning and optimizations we can improve the

predictive ability of our model using evidential methods, which will improve the reported IoU.

The next step in our future work would consist of training the active learning level on the highly accurate instances generated by an expert annotator in the fast supervised cloud mask generation process. Using this dataset and some modifications/optimizations to the network, as can be extrapolated from this thesis' work, we expect the result to be the realized ability to generate large and accurate cloud mask datasets.

Even though this work contributes a modified network to be the backbone of the active learning loop, we would consider using network architectures superior to ENet in the image segmentation domain. One such architecture, DeepLabv3+ [11], shows much promise from the standpoint of pure segmentation accuracy performance. DeepLabv3+ was developed to resolve the problem of segmenting objects at multiple scales. The authors built modules that employ atrous convolution in cascade or in parallel to capture multi-scale context as they adopt various atrous rates. Ultimately, as a network superior to ENet in segmentation tasks, using DeepLabv3+ seeks to train a model to output cloud masks at even higher accuracy than ENet.

Using something like DeepLabv3+ would make the system slightly more heavyweight. Contrarily, we also consider consolidate this two level system for use in an on-line setting. This effort would include retaining use of ENet or other lightweight architectures and employing other image processing optimizations, classifier optimizations, and parallelization to increase the throughput of the system.

Lastly, in this thesis, when developing and testing the fast supervised cloud mask generation process, using the Sentinel-2 LC1 product, we found that using solely the RGB channels is insufficient to extend the described annotation process to satellite images with fewer channels, e.g. Planetscope images. To tackle this, we planned to involve more spatial information and a local textural analysis. We attempted this via our experimentation with patches, however this proved unsatisfactory as well. Next steps would involve other algorithms that take textural information into account, in addition to potentially the development of new spectral indices (linear or non-linear

combinations of existing spectral bands).

Bibliography

[1] National Highway Traffic Safety Administration. Tesla Crash Preliminary Evaluation Report. Technical report, U.S. Department of Transportation, 01 2017.

[2] Selim Aksoy, Krzysztof Koperski, Carsten Tusk, Giovanni Marchisio, and James Tilton. Learning bayesian classifiers for scene classification with a visual grammar. *Geoscience and Remote Sensing, IEEE Transactions on*, 43:581 – 589, 04 2005.

[3] Alexander Amini, Ava Soleimany, Sertac Karaman, and Daniela Rus. Spatial uncertainty sampling for end-to-end control, 2019.

[4] Dana Angluin. Queries revisited. *Theoretical Computer Science*, 313(2):175 – 194, 2004. Algorithmic Learning Theory.

[5] Aydin Ayanzadeh. A study review: Semantic segmentation with deep neural networks, 03 2019.

[6] L. Baetens, C. Desjardins, and O. Hagolle. Validation of copernicus sentinel-2 cloud masks obtained from maja, sen2cor, and fmask processors using reference cloud masks generated with a supervised active learning procedure. *Remote Sensing*, 11, 02 2019.

[7] Louis Baetens, Camille Desjardins, and Olivier Hagolle. Validation of copernicus sentinel-2 cloud masks obtained from maja, sen2cor, and fmask processors using reference cloud masks generated with a supervised active learning procedure. *Remote Sensing*, 11, 02 2019.

[8] Louis Baetens and Olivier Hagolle. Sentinel-2 reference cloud masks generated by an active learning method, October 2018.

[9] S. Bandyopadhyay, U. Maulik, and A. Mukhopadhyay. Multiobjective genetic clustering for pixel classification in remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 45(5):1506–1511, 2007.

[10] Cenk Baykal, Lucas Liebenwein, Dan Feldman, and Daniela Rus. Low-regret active learning. *CoRR*, abs/2104.02822, 2021.

[11] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018.

[12] Roberto Cilli, Alfonso Monaco, Nicola Amoroso, Andrea Tateo, Sabina Tangaro, and Roberto Bellotti. Machine learning for cloud detection of globally distributed sentinel-2 images. *Remote Sensing*, 12:2355, 07 2020.

[13] T. Dagobert, R. Grompone von Gioi, C. de Franchis, J.-M. Morel, and C. Hessel. Cloud Detection by Luminance and Inter-band Parallax Analysis for Pushbroom Satellite Imagers. *Image Processing On Line*, 10:167–190, 2020.

[14] T. Dagobert, R. Grompone von Gioi, J.M. Morel, and C. de Franchis. Temporal Repetition Detector for Time Series of Spectrally Limited Satellite Imagers. *Image Processing On Line*, 10:62–77, 2020.

[15] A. Dawid. The well-calibrated bayesian. *Journal of the American Statistical Association*, 77:605–610, 1982.

[16] M. Degroot and S. Fienberg. The comparison and evaluation of forecasters. *The Statistician*, 32:12–22, 1983.

[17] M. Drusch, U. Del Bello, S. Carlier, O. Colin, V. Fernandez, F. Gascon, B. Hoersch, C. Isola, P. Laberinti, P. Martimort, A. Meygret, F. Spoto, O. Sy, F. Marchese, and P. Bargellini. Sentinel-2: Esa's optical high-resolution mission for gmes operational services. *Remote Sensing of Environment*, 120:25–36, 2012. The Sentinel Missions - New Opportunities for Science.

[18] Steve Foga, Pat L. Scaramuzza, Song Guo, Zhe Zhu, Ronald D. Dilley, Tim Beckmann, Gail L. Schmidt, John L. Dwyer, M. Joseph Hughes, and Brady Laue. Cloud detection algorithm comparison and validation for operational landsat data products. *Remote Sensing of Environment*, 194:379–390, 2017.

[19] Linton C. Freeman. *Elementary applied statistics: for students in behavorial science*. Wiley, 1965.

[20] M.A. Friedl and C.E. Brodley. Decision tree classification of land cover from remotely sensed data. *Remote Sensing of Environment*, 61(3):399–409, 1997.

[21] Yarin Gal and Zoubin Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference, 2016.

[22] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning, 2016.

[23] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian active learning with image data. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1183–1192, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[24] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. *CoRR*, abs/1703.02910, 2017.

[25] Yonatan Geifman and Ran El-Yaniv. Deep active learning over the long tail. *CoRR*, abs/1711.00941, 2017.

[26] Alex Goupilleau, Tugdual Ceillier, and Marie-Caroline Corbineau. Active learning for object detection in high-resolution satellite images, 2021.

[27] Alex Graves. Practical variational inference for neural networks. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[28] R. Grompone von Gioi, C. Hessel, T. Dagobert, J.-M. Morel, and C. de Franchis. Temporal repetition detection for ground visibility assessment. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2:829–835, 2020.

[29] Yanming Guo, Yu Liu, Theodoros Georgiou, and Michael Lew. A review of semantic segmentation using deep neural networks. *International Journal of Multimedia Information Retrieval*, 7, 06 2018.

[30] Jessica Guynn. Google photos labeled black people 'gorillas', Jul 2015.

[31] O. Hagolle, M. Huc, D. Villa Pascual, and G. Dedieu. A multi-temporal method for cloud detection, applied to formosat-2, venμs, landsat and sentinel-2 images. *Remote Sensing of Environment*, 114(8):1747–1755, 2010.

[32] Min Han, Xinrong Zhu, and Wei Yao. Remote sensing image classification based on neural network ensemble algorithm. *Neurocomputing*, 78:133–138, 02 2012.

[33] Elmar Haussmann, Michele Fenzi, Kashyap Chitta, Jan Ivanecky, Hanson Xu, Donna Roy, Akshita Mittel, Nicolas Koumchatzky, Clement Farabet, and Jose M. Alvarez. Scalable active learning for object detection, 2020.

[34] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018.

[35] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *CoRR*, abs/1610.02136, 2016.

[36] Matt Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference, 2013.

[37] André Hollstein, Karl Segl, Luis Guanter, Maximilian Brell, and Marta Enesco. Ready-to-use methods for the detection of clouds, cirrus, snow, shadow, water and clear sky pixels in Sentinel-2 MSI images. *Remote Sensing*, 8(8):1–18, 2016.

[38] Lang K. and E. Baum. Query learning can work poorly when a human oracle is used. pages 335–340. IEEE International Joint Conference on Neural Networks, IEEE Pres, 1992.

[39] G. Kaplan and U. Avdan. Object-based water body extraction model using sentinel-2 satellite imagery. *European Journal of Remote Sensing*, 50:137–143, 03 2017.

[40] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision?, 2017.

[41] Julia Kho. Why random forest is my favorite machine learning model, Mar 2019.

[42] Ross King, Jem Rowland, Stephen Oliver, Meong Young, Wayne Aubrey, Emma Byrne, Maria Liakata, Magdalena Markham, Pınar Pir, Larisa Soldatova, Andrew Sparkes, Ken Whelan, and Amanda Clare. The automation of science. *Science (New York, N.Y.)*, 324:85–9, 05 2009.

[43] Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. Learning active learning from real and synthetic data. *CoRR*, abs/1703.03365, 2017.

[44] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.

[45] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2017.

[46] Cheng Chien Liu, Yu Cheng Zhang, Pei Yin Chen, Chien Chih Lai, Yi Hsin Chen, Ji Hong Cheng, and Ming Hsun Ko. Clouds classification from Sentinel-2 imagery with deep residual learning and semantic image segmentation. *Remote Sensing*, 11(2), 2019.

[47] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.

[48] Christos Louizos and Max Welling. Structured and efficient variational deep learning with matrix gaussian posteriors, 2016.

[49] David Mackay, John Bridle, Peter Cheeseman, Sidney Fels, Steve Gull, Andreas Herz, John Hopfield, Doug Kerns, Allen Knutsen, David Koerner, Mike Lewicki, Tom Loredo, Steve Luttrell, Ken Rose, Sibusiso Sibisi, John Skilling, Haim Sompolinsky, and Nick Weir. Bayesian methods for adaptive models. 05 1991.

[50] Ramona Magno, Leandro Rocchi, Riccardo Dainelli, Alessandro Matese, Salvatore Di Gennaro, Chi-Farn Chen, Nguyen thanh son, and Piero Toscano. Agroshadow: A new sentinel-2 cloud shadow detection tool for precision agriculture. *Remote Sensing*, 13:1219, 03 2021.

[51] M. Main-Knorn, B. Pflug, J. Louis, V. Debaecker, U. Müller-Wilm, and F. Gascon. Sen2cor for sentinel-2. page 3, 10 2017.

[52] K. F. Manizade, J. D. Spinhirne, and R. S. Lancaster. Stereo cloud heights from multispectral infraredider imagery via region-of-interest segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 44(9):2481–2491, Sept 2006.

[53] Giorgos Mountrakis, Jungho Im, and Caesar Ogole. Support vector machines in remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(3):247–259, 2011.

[54] Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, Berlin, Heidelberg, 1996.

[55] Arthur Ouaknine. Review of deep learning algorithms for image semantic segmentation, Oct 2020.

[56] John Paisley, David Blei, and Michael Jordan. Variational bayesian inference with stochastic search, 2012.

[57] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation, 2016.

[58] EO Research. Cloud masks at your service, Jul 2020.

[59] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[60] Alber Hamersson Sanchez, Michelle Cristina A. Picoli, Gilberto Camara, Pedro R. Andrade, Michel Eustaquio D. Chaves, Sarah Lechler, Anderson R. Soares, Rennan F. B. Marujo, Rolf Ezequiel O. Simões, Karine R. Ferreira, and Gilberto R. Queiroz. Comparison of cloud cover detection algorithms on sentinel–2 images of the amazon tropical forest. *Remote Sensing*, 12(8), 2020.

[61] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach, 2018.

[62] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty, 2018.

[63] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[64] D. Shin and J. K. Pollard. Cloud height determination from satellite stereo images. In *IEE Colloquium on Image Processing for Remote Sensing*, pages 4/1–4/7, Feb 1996.

[65] Ava P Soleimany, Alexander Amini, Samuel Goldman, Daniela Rus, Sangeet N Bhatia, and Connor W Coley. Evidential deep learning for guided molecular property prediction and discovery. 2020.

[66] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015.

[67] F. Wang. Fuzzy supervised classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 28:194–201, 1990.

[68] Gui-Song Xia, Zifeng Wang, Caiming Xiong, and Liangpei Zhang. Accurate annotation of remote sensing images via spectral active clustering with little expert knowledge. *Remote Sensing*, 7:15014–15045, 11 2015.

[69] Zhe Zhu, Shixiong Wang, and Curtis E. Woodcock. Improvement and expansion of the fmask algorithm: cloud, cloud shadow, and snow detection for landsats 4–7, 8, and sentinel 2 images. *Remote Sensing of Environment*, 159:269–277, 2015.

[70] Anze Zupanc. Improving cloud detection with machine learning, Jul 2020.