Image Segmentation Stability: An Empirical Investigation

School of Computing, University of South Alabama

Abstract—This paper is an evaluation of the stability of image segmentation algorithms. We provide a definition of what it means for a segmentation algorithm to be stable, as well as a procedure for formulating stability measures in a given context. Using this procedure, a systematic empirical analysis of the stability of three popular image segmentation algorithms is performed. The stability of the algorithms is examined in the context of three real world transformations: video, compression and subimage alterations. The analysis provides evidence that stability is a meaningful property of an algorithm for a given set of parameters in a given application, and could be helpful for performing model and hyperparameter selection. This result lays the groundwork for developing methods for using stability as a means of automatic model and hyperparameter selection.

Index Terms—image segmentation, superpixels, stability

I. INTRODUCTION

Model validation and hyperparameter selection for unsupervised learning algorithms are not easy problems. Supervised algorithms have a natural procedure for hyperparameter selection: divide the data into test and training sets and choose the model that performs best against the test sets [1]. Unsupervised algorithms have no such procedure, and metrics for measuring the quality of a result are often ad hoc and limited.

Image segmentation and superpixelation algorithms are no exception to this rule. An image segmentation algorithm is a specific type of clustering algorithm concerned with clustering image pixels into semantically related groups; e.g. ‘sky’ vs. ‘sea’ vs. ‘tree’. As such, models can be validated with metrics of quality that are available to clustering algorithms, such as compactness, but also image specific metrics such as explained color variation [2]. However, these metrics only capture a limited view of what makes a ‘good’ image segmentation. Compactness penalizes correct segmentations of oddly-shaped objects, and explained color variation penalizes segments with variations in color, even if they have consistent texture [3].

In addition to these metrics, segmentation and superpixelation algorithms are frequently benchmarked against a single ground-truth dataset, the Berkeley Segmentation Dataset and Benchmark (BSDS500) [4]. BSDS500 consists of a set of images that have been hand-segmented to provide a means of comparison of an automated segmentation against a human one. However, optimal models or hyperparameters for the BSDS500 dataset, which is approximately 50% images of ‘nature’ scenes, may not transfer to other applications, such as facial recognition. Furthermore, the hyperparameters of some algorithms are affected by image meta-information. For example, many superpixel algorithms require the user to pick the number of superpixels, which depends on resolution. Every image in the BSDS500 dataset is $481 \times 321$. This is convenient for comparing the results between images in the dataset, but could pose problems when applying optimal BSDS500 hyperparameters to other image sets. With this in mind, it would be incredibly useful to have a means of model and hyperparameter selection that does not depend on a ground-truth dataset, that does not depend on limited a priori expectations of the qualities of a ‘good’ segmentation, and that is appropriate across datasets and applications.

Recently, the stability of a clustering has become a popular metric to aid parameter selection of a model, and to validate the results [5]. The idea is that a ‘correct’ clustering should be robust to small-scale perturbations in individual data points, because large-scale structure will be retained. Instability of a clustering may therefore indicate problems with the clustering, or with the clustering algorithm. The benefit of using stability as a measure of quality is that it is model-independent and does not require application-specific qualities of clustering ‘goodness’. Indeed, for some algorithms, theoretical results exist showing that the ‘true’ clustering is a stable one [6]. However, it is not foolproof. For example, an algorithm that always returns the same clustering regardless of input data has perfect stability. Even so, stability has been used successfully to measure the quality of a clustering [7] and also to optimize model hyperparameters [5].

Image segmentation and superpixelation is to the field of computer vision as clustering is to statistics and machine learning. It is one of the oldest and most studied problems in computer vision, and image segmentation algorithms are frequently used as a pre-processing step for other applications [8]. Areas that make use of image segmentations as a matter of course include image retrieval [9] medical imaging [10] and object detection [11]. It is therefore important to have objective metrics to evaluate an image segmentation algorithm, or a particular segmentation. For example, an image retrieval system would benefit from algorithms that avoid finding patterns in the addition of random noise due to image quality issues and non-random noise due to compression. An algorithm used in a motion-tracking system should ensure the retention of found patterns under motion (video) through time. An image tampering detection system would require an algorithm that retains found patterns under discrete changes, such as cropping.

Since stability has been successful in evaluating clustering algorithms, and image segmentation is such a closely related
field, we hypothesize that stability will be a useful measure for model and hyperparameter selection of image segmentation algorithms. With this in mind, in the context of several real-world scenarios, the remainder of this paper examines stability maximization as a potential procedure for selecting amongst various algorithms and hyperparameters. Section 2 presents related works. Section 3 proposes a general framework for generating measures of stability corresponding to a particular class of transformation and a particular comparison measure. Section 4 lays out the experimental setup and our reasonings behind the classes of image perturbations, comparison measures and algorithms chosen for evaluation. Section 5 evaluates the stability of the selected algorithms for each comparison measure and class of perturbation, discussing possible reasons why certain algorithms may be more/less stable than others. Section 6 discusses potential application areas where the current analysis may be relevant, such as situations where there may be a trade-off between having a ground-truth-accurate algorithm and a transformation-stable algorithm. Finally, the conclusion summarizes lessons learned and opportunities to continue the current research or to extend it to related fields.

II. RELATED WORKS

Image segmentation algorithms are closely related to the general class of clustering algorithms. There is a reasonably large body of work about measuring the stability of various clustering algorithms to changes in the underlying dataset. There is also much work in the creation of clustering algorithms that are specifically designed to display some degree of stability. This section describes some of the work from this area that may be relevant to the subject of image segmentation stability. Furthermore, we identify some topics in image segmentation which are related to stability. Finally, we discuss some results for image segmentation specifically, and summarize how the present work differs.

A related problem addresses the statistical consistency of various clustering algorithms. A consistent estimator is one for which, as the size of the sample set is increased, the estimator converges to the “true” population value. In the case of clustering algorithms, evidence of consistency might exhibit as stability of the algorithm to the addition of new sample data from the same population. Theoretical results are available for a select few algorithms, such as k-means [12] and spectral clustering [13], but such results are not known or well-studied for most algorithms. However, since real-world images form an extremely narrow segment of the set of all possible images, an algorithm that is empirically consistent for a given application may be sufficient for most use cases. Either way, the problem of consistency in the context of image segmentation relates to stability under changes in resolution, which has not been studied to our knowledge, either theoretically or empirically.

The stability of general clustering algorithms has been studied in [5] [14]. Much of the work is confined to a small subset of clustering algorithms and comparison methods. In [6] clustering stability of the k-means algorithm is used as a way to select an appropriate k. In [15] clustering stability of hierarchical clustering algorithms using the Goodman-Kruskal gamma statistic (which is specific to hierarchical clusterings) is used as a measure of the validity of a given clustering. In [7] the stability of k-means and hierarchical clustering algorithms with respect to the alteration of data in a meaningful but non-random fashion, such as might happen in time-series.

Incremental clustering algorithms are a class of algorithms which inherently account for stability over time-series changes in the training dataset. The task here is to efficiently update the previously generated clusters due to the arrival of new data in a data streaming model [16] [17]. New data are generated independently from some underlying statistical distribution and the model is updated accordingly. In this context the previous data are still relevant to the desired clustering, and are explicitly included in the parameter estimation of the new clustering. Although most image segmentation algorithms are designed for static image analysis, there has been some research on the creation of algorithms capable of incremental updates [18]. For both image segmentation algorithms and general clustering algorithms, temporal stability under incremental updates is an important characteristic [19], and a thorough investigation is probably warranted.

Further work along a similar vein is sometimes referred to as “evolutionary” clustering, i.e. clusterings that change over time due to changes in the underlying dataset. These works seek to explicitly enforce certain characteristics, including temporal stability, on existing clustering algorithms. In contrast to incremental clustering, instead of the previous data being carried over into the current model, the previous model parameters are incorporated into the formula for the present ones. A common application area for such algorithms is in social-network analysis, where points and edges change over time and prior states of the network provide relevant information to the current state, but are no longer valid data points. In [20] k-means and hierarchical clustering algorithms are modified to incorporate requirements on temporal stability. In [21], this idea is extended to spectral clustering methods. [22] applies temporal constraints to Dirichlet process models in order to address some shortcomings in the previous works such as a requirement for constant cluster sizes. These works are also limited in the number of algorithms that they evaluate, possibly due to the difficulty of imposing temporal restrictions on arbitrary algorithms. These methods are inherently stable by design, but these considerations are not taken into account in most work on image segmentation algorithms.

Financial time-series clustering is another application area in which temporal stability of clustering algorithms has been found to be of consequence. The use case here is one of
individual data points being time-series, and the clustering algorithm is tasked with grouping time-series according to some similarity metric [23]. It is important in this context that an algorithm give time stable clusterings to prevent completely upending an asset portfolio based on minor fluctuations in the market. In [24] a framework is proposed for evaluating the stability of time-series clusterings according to a wide variety of criteria, including temporal stability. However, to the best of our knowledge, no attempt has been made to translate the rigor of these stability measurements in financial analysis to the realm of image segmentation.

Another problem that is superficially similar to the task at hand is the generation of spatio-temporal clusterings [25] [26]. In the context of image segmentation this amounts to video segmentation such as with supervoxel algorithms [27]. The goal here is to generate clusterings over a point set that has both spatial dimensions as well as a temporal dimension. The problem of clustering stability is moot for these applications, since separate clusterings are not generated for different time points, but a single clustering is generated that includes time as a dimension in the individual data.

Specifically with respect to image segmentation and superpixels, there has been some research in evaluating the stability of algorithms over small changes to the input. However, algorithm comparisons have largely been confined to comparison of clusterings with ground-truth datasets, such as the Berkeley Segmentation Dataset and Benchmark [4] and the PASCAL VOC challenge dataset [28] and a priori measures of clustering quality such as compactness and color variation [2]. In [29] an evaluation is made of the stability of various segmentation algorithms under affine transformations. This is important primarily for video applications, since a change in camera angle of a flat stationary object will be observed as an affine transformation from the user’s viewpoint. However, this only covers a small proportion of possible image perturbations. Neubert and Protzel [30] propose a measure for evaluating the temporal stability of an image segmentation algorithm, and evaluate a number of algorithms on the KITTI [31] and Sintel [32] optical flow datasets. However, the measure introduced in this work is specifically based on undersegmentation error, which does not penalize the number of clusters. Also, their stability measure is only applicable in the context of continuous-motion transformations in video, and requires ground-truth data.

This paper adds to the depth of previous explorations of image segmentation algorithm stability by measuring stability in contexts other than video, and in the absence of ground-truth. In addition, it extends the definition in [5] of the stability of a clustering algorithm over random perturbations to include other classes of transformations. This definition is used to devise stability measures for a segmentation algorithm under transformations from video, compression and sub-image alteration. To our knowledge, a systematic study of the stability of image segmentation algorithms to these types of transformations, and others, in the absence of ground-truth, has not been investigated until now.

III. STABILITY

We generalize to clustering algorithms taking arbitrary parameters a procedure given in [5] for evaluating the instability of a clustering algorithm. Note that this procedure measures the instability of a clustering algorithm, since the distance increases as two clusterings diverge. Given some domain \( D \) from which data is acquired, a clustering algorithm \( A \) with parameters \( \theta \) that produces clusterings \( C : D \rightarrow \mathbb{N} \), a means of generating a dataset \( S \) and a perturbed version of that dataset \( S' \) (e.g., by subsampling both from a larger dataset, by adding noise to a given \( S \), by applying some arbitrary transformation to \( S \), etc) and a comparison (distance or similarity) measure \( d \) between two clusterings:

1. Generate two datasets \( S \) and \( S' \).
2. Cluster the data sets \( S, S' \) with algorithm \( A \) to obtain clusterings \( C, C' \).
3. Compute the comparison measure \( d(C, C') \) between the clusterings.

The stability of the algorithm \( A \) can be seen as a property measurable from these distances computed at various points \( S, S' \). In general, it will be useful to iterate this procedure over multiple perturbations and compute some summarizing statistic of the resulting set of distances. For example, in [5], the \( S, S' \) are generated by subsampling from some given dataset, and a mean is computed for the generated set of pairwise distances. This summarizing statistic can be used to compare the performance of two algorithms \( A, A' \). It can also potentially be minimized over the algorithm parameters \( \theta \) to obtain the ‘best’ parameters.

This concept of stability mirrors the concept of function continuity, i.e. a clustering algorithm \( A \) is stable if small changes to the input \( S, S' \) produce small changes in the output \( C, C' \). In general, it will be useful to iterate this procedure over multiple perturbations and compute some summarizing statistic of the resulting set of distances. For example, in [5], the \( S, S' \) are generated by subsampling from some given dataset, and a mean is computed for the generated set of pairwise distances. In the proceeding section, we will be using the mean and standard deviation as useful summarizing statistics of stability.

Note that the means by which one generates the perturbed inputs \( S, S' \) is important. For example, an image segmentation algorithm may be stable under continuous motion, but not stable under compression. Such an algorithm might be more appropriate for the task of tracking objects in a video stream that for performing image search on a database of compressed images. With this in mind, our algorithm comparison involves several different transformations on different datasets. Note also that there are a large number of possible distance measures that are available between two clusterings. Most of these distance measures involve some sort of operation on the contingency table, essentially counting points that clusters share or do not share. Some, however, do not, and are permissive of certain relationships between clusterings, such as refinement. In our evaluation, we will be using a handful of
measures popularly used for image segmentation comparison to cover a variety of qualities.

IV. METHODOLOGY

The experiments outlined here are an exploratory investigation into the properties of image segmentation algorithm stability. Establishing an actual procedure for model validation or hyperparameter selection is beyond the scope of this paper, although such applications have been shown in the literature for general clustering algorithms. This work will focus instead on establishing the foundations of such a study; whether it is meaningful to talk about stability as a property of an image segmentation algorithm, instead of simply between two individual clusterings; whether stability is, in part, dependent on the perturbation that is being applied, and is thus not a one-size-fits-all measure; and whether some algorithms or hyperparameter selections are inherently more stable in a particular context than others.

A. Algorithms

The Experiments section will be concerned with evaluating the stability of the following image segmentation algorithms. First, we consider the graph-based method by Felzenswalb and Huttenlocher (FH) [33]. FH begins by constructing a weighted graph with edges between neighboring pixels. Each pixel starts in its own segment. It proceeds to greedily combine segments when the internal variation of both of the components is less than the variation along a point of their boundary. The internal variation is defined as the maximum weighted edge in the minimum spanning tree of the component plus some user-defined parameter \(\tau\). Without \(\tau\), FH can have a tendency to create very small segments, since internal variation can be swayed by a single pixel’s relationship to its neighbors. For us, \(\tau\) is chosen for a component \(C\) using the example in their paper (Eq. 1), where \(k\) is a constant hyperparameter of the algorithm.

\[
\tau(C) = k/|C|
\]

In addition the image is pre-processed with a Gaussian filter per the suggestion in the original paper, and \(\sigma\) for the filter is considered to be a hyperparameter of the algorithm.

Second, we consider the Simple Linear Iterative Clustering (SLIC) algorithm [34]. SLIC performs \(k\)-means clustering in the 5D space formed by concatenating the CIELAB color and spatial information for a pixel. When combining the two, it is notable that distance in color space and distance in pixel position space have different meanings and extents. Thus, for a distance measure in the combined space SLIC introduces a scaling parameter \(m\) that controls the weight of pixel position space distances. \(m\) has the effect of controlling the compactness of clusters by (de)emphasizing the importance of pixel positions relative to cluster centers. \(k\) is the number of segments, analogous to \(k\)-means. Per the suggestion in the original paper, the algorithm is run for a maximum of 10 iterations. Since FH does not explicitly allow for control of the number of segments, to allow for comparison, we chose \(k\) to be the approximate number of segments found by FH for a data sequence.

Finally, we consider the Superpixels Extracted via Energy-Driven Sampling (SEEDS) algorithm [35]. SEEDS begins by dividing the image into a regular grid. SEEDS then operates by swapping groups of pixels between segments, hill-climbing to a local optimum of an image-wide objective function. The objective function involves two terms. One penalizes the \(L_2\) norm of the superpixel color histograms. The other penalizes the \(L_2\) norm of the superpixel label histogram for boundary pixel neighborhoods. The size of the \(N \times N\) neighborhoods to use for computing boundary pixel histograms is a hyperparameter of the algorithm, as well as \(\gamma\), the relative weight of the two terms, and the number of desired superpixels \(s\). We use the defaults of \(3 \times 3\) neighborhoods and \(\gamma = 1\) suggested in the original paper, and implemented in the publicly available code. \(s\) is chosen using the same procedure used for SLIC.

B. Datasets

The stability of the previously mentioned algorithms are examined with respect to the following perturbations. First, we study stability under sub-image alterations. This is achieved using a dataset of hard-disk images for which image segmentation is used to discover file boundaries. Drive contents are gathered by creating a stock Ubuntu 14.04 virtual machine installation, running a general system update, and then incrementally installing a variety of packages. The exact packages installed are detailed in Table ???. Images are then created using an established operation similar to bayeplot [36] for the visualization of drive contents. Before each operation, each byte of the drive is mapped into a color space \([(\text{How? Ask Jordan})]\), and the resulting color string is converted to 2D by taking fixed-length windows of the string and stacking them. In the resulting image, files and related sub-file sectors tend to have similar colors and/or textures (Fig. ??). Image analysis techniques can then be applied to the resulting images for e.g. malware detection [37]. Since a file update will generally only affect a small section of the drive, instability in this context amounts to the addition of a file in one area of the drive affecting the computed segmentation elsewhere in the image. Stability is measured as the distance between segmentations on neighboring stages of the update process.

Second, we study stability of a segmentation under continuous non-random changes with scenes from video data. This is achieved using inter-frame stability on the first still-camera video from the Video Image Retrieval and Analysis Tool (VIRAT) ground dataset (VIRAT_S_000001.mp4) [38]. Videos are taken apart at a rate of 5 frames per second. Stability is measured as the distance between segmentations on neighboring extracted frames. Since the videos are compressed, segmentation algorithms will produce different segmentations between two neighboring frames even in the absence of motion. In order to account for this, we specifically analyze a section of a video with no motion and compare to the remainder of the video, and also provide an analysis of stability under the effects of compression. This video was...
chosen simply because it was first. More were not evaluated to eliminate bias in the results, because sufficient insights were gained from this video to continue the experiments and also due to the fact that segmenting 5 frames per second of this video takes about 12 hours on a single i7-4700MQ core.

Third, we study stability under compression to an increasingly lossy format. For this, we use image ‘175083.jpg’ from the Berkeley Segmentation Dataset (BSDS300) [4]. Images are compressed in a jpeg format incrementally in 50 steps to the point of being almost entirely noise. We measure the stability as the distance between segmentations on neighboring levels of compression. This image was chosen because it featured prominently in [2]. Again, more were not evaluated to eliminate bias and because of the intense resource usage of repeated application of segmentation algorithms.

C. Metrics

For each combination of algorithm and perturbation, we use the following comparison measures in our stability calculations. Variation of Information (VOI) has pleasing theoretical roots and is a true metric [39] (Eq. 2). Here \( H \) is Shannon entropy and \( I \) is mutual information.

\[
\text{VOI}(C, C') := H(C) + H(C') - 2I(C, C') \tag{2}
\]

The intuition behind VOI is that we can compute the information given by a clustering \( C \) that is over and above the information given by \( C' \) by subtracting their mutual information from the entropy of \( C \). That quantity can be made symmetric by computing the same quantity for \( C' \) and adding the two values. The result is a quantification of the information the two clusterings contain that they do not both contain.

The Rand Index (RI) is the unsupervised equivalent of accuracy [40] (Eq. 3). Here \( n_{11} \) is the number of pairs of points that lie in the same cluster in both \( C \) and \( C' \), \( n_{00} \) are the number of pairs of points that lie in a different cluster in \( C \) and \( C' \), and \( n \) is the number of points.

\[
\text{RI}(C, C') := \frac{n_{00} + n_{11}}{\binom{n}{2}} \tag{3}
\]

The intuition behind RI is that, \( C' \) is ‘correct’ according to \( C \) exactly when a pair of points that are in the same cluster in \( C \) are in the same cluster in \( C' \), or when a pair of points that are in different clusters in \( C \) are in different clusters in \( C' \). The ratio of ‘correct’ pairs to all possible pairs results in the RI. Both VOI and RI can be computed by some operation on the contingency table, and can thus be viewed as summarizing statistics of the contingency table. In this sense, they are both measures of the degree to which two clusterings agree or disagree, weighting all points equally.

Local Consistency Error (LCE) is a measure of the degree to which one segmentation is a refinement of the other, allowing for the refinement to be in different directions in different parts of the image [4] (Eq. 5): Here \( R(C, p_i) \) is the segment corresponding to pixel \( p_i \), and \( n \) is the number of pixels.

\[
E(C, C', p_i) := \frac{|R(C, p_i) - R(C', p_i)|}{|R(C, p_i)|} \tag{4}
\]

\[
\text{LCE}(C, C') := \frac{1}{n} \min \{ E(C, C', p_i), E(C', C, p_i) \} \tag{5}
\]

Note that \( \text{LCE}(C, C') \) will be exactly zero when the segments of \( C \) are subsegments of segments of \( C' \) or vice versa, or some mixture of those two scenarios. This is a useful metric because image segmentations are frequently used as a preprocessing step for further analysis, and ‘oversegmentation’, having more segments than strictly necessary, is an acceptable situation.

Boundary Displacement Error (BDE) is a measure of the degree to which the boundary pixels of two segmentations align [41] (Eq. 6). Here \( B(C) \) is the set of boundary pixels in \( C \) (neighboring pixels that reside in different clusters).

\[
\text{BDE}(C, C') := \sum_{p \in B(C)} ||p, B(C')||_2 \tag{6}
\]

Whereas the previous metrics weight all misclustered pixels equally, BDE weights pixels based on how close they are to the edge of a cluster. Like LCE, BDE is permissive of oversegmentations, but only in one direction.

V. EXPERIMENTS

In this section we detail the results of our experiments. The experiments proceed according to the aforementioned goal of determining the feasibility of using stability as a means of model and hyperparameter selection. Results are given chronologically, as the experiment progresses, since results from a particular analysis inform the process of proceeding experiments. Where particular examples are used, they are chosen arbitrarily, and not cherry-picked to give ‘good’ results.

In the Algorithm Comparison section, for each combination of dataset/perturbation procedure, distance measure and algorithm, we measure the stability as the sequence of distances between the perturbed images. Mean and standard deviation are evaluated as potential summarizing statistics to measure the ‘difference’ between the stability of various algorithms.

In the Hyperparameter Comparison section, we measure the VOI stability of FH using a variety of hyperparameters. In the absence of any fixed objective function, the mean and standard deviation of the sequence of distances are evaluated in tandem, loosely, to find the ‘most’ stable solution. Because of the lack of a concrete objective function, a manual search is performed. Starting with a grid, hyperparameter combinations are evaluated in-between or outside of the grid points as they subjectively appear that they may offer a lower mean or standard deviation, or lend evidence to the existence of a local minimum at a previously evaluated point. For all of the experiments performed, a discussion is included regarding the process of the investigation, and lessons learned.

A. Algorithm Comparison

The results of the first set of experiment results are illustrated in Figs. 1, 2 and 3. These graphs are essentially a time series of the distances between segmentations of successive images. A few features of these graphs are worth noting.

One feature of note from every dataset is that although VOI, LCE and BDE tend to at least be of the same order
of magnitude, RI is totally different. This can possibly be attributed to the variable number of segments given by FH, and the outsized effect of the number of segments on RI [42]. If stability with respect to the number of segments is a concern, RI might be useful. Otherwise, a better measure for stability of algorithms with variable numbers of clusters might be the Adjusted Rand Index [43], which is more robust to number of segments. Owing to this problem, RI is generally left out of proceeding discussions.

Another feature that is apparent from all of the datasets (Figs. 1, 2 and 3) is that the distances between successive frames for the SEEDS model are reliably greater than corresponding distances for FH, with the only exception being BDE for the compression dataset. The existence of a logical pattern to these values suggests that stability is a meaningful property of an algorithm for a given set of hyperparameters and a distance measure. This is opposed to having seen noisy values that were sometimes larger for one algorithm, and sometimes larger for another with little or no rhyme or reason. I.e. In a given context, it is valid to talk about a model being ‘more’ stable or ‘less’ stable than another.

Lastly, these graphs suggest that the mean distance between successive images is, by itself, an insufficient summary of the stability of a model. This can be seen in several of the graphs, but appears most strikingly in the graph of VOI for the video dataset. First, it is important to realize that the successive images in this dataset are actually different images, so some level of difference between the segmentations of successive images is ‘true’, and should be expected. Since the ‘true’ segmentations are ill-defined, the ‘true’ amount of variation between them is as well, and is thus impossible to measure in practice. Nevertheless, the amount of motion over 200ms in the example video is small, and never abrupt (in fact, the first minute and a half involve no motion at all, the only differences being compression artifacts). Thus, it is expected that the ‘true’ segmentations from image 1 to 2 to 3 etc. have an approximately constant distance between them. This means that variability is also an important summarizing statistic of model stability. Note from VOI in Fig. 1 that although the mean distance between frames for FH and SLIC are similar ($\bar{x}_{FH} \approx 0.54$ and $\bar{x}_{SLIC} \approx 0.49$), FH generates similar segmentations for video frames much less reliably ($s_{FH} \approx 0.19$, $s_{SLIC} \approx 0.06$). One possible reason for this disparity is that, because FH is a greedy algorithm, initial conditions have a larger effect on the final outcome. Either way, it is clear that the mean distance does not tell the whole story of stability in practice.

B. Hyperparameter Comparison

This section investigates the effects of hyperparameter selection on algorithm stability. This is performed with the video dataset using the FH algorithm and VOI distance. The results of this second set of experiments are illustrated in Figs. 6, 7, intended to illustrate the effect of hyperparameter $k$, and ?? to illustrate the effect of $\sigma$. The actual segmentations at several points are provided corresponding to the hyperparameters under which the algorithm shows relatively high or low stability in Fig. 8. For comparison, the segmentation using $\sigma = 0.8$ and $k = 300$ from the original paper is also provided.

As can be seen in Figs. 6 and 7, the stability of the FH algorithm with respect to choice of $k$ follows a general pattern. The algorithm is highly unstable for low $k$, exhibits at least one local minimum, peaks, and then drops toward 0 as $k$ gets very large.
Fig. 3. Distances between the segmentations of neighboring updates in the drive contents dataset.

Fig. 4. The mean VOI between segmentations of neighboring frames of the first VIRAT video for various FH hyperparameters.

Fig. 5. The standard deviation of VOI between segmentations of neighboring frames of the first VIRAT video for various FH hyperparameters.

Fig. 6. The mean VOI between segmentations of neighboring frames of the first VIRAT video for FH for various σ. The first frame of this video can be seen in 8

REFERENCES


Fig. 7. Segmentations of the first frame of the first VIRAT video for FH for various $\sigma$.


