Exploring Video Frame Redundancies for
Efficient Data Sampling and Annotation in Instance Segmentation

Jihun Yoon, Min-Kook Choi*
AI Dev. Group, hutom
Seoul, Republic of Korea
{yjh2020, mkchoi}@hutom.io

Abstract

In recent years, deep neural network architectures and learning algorithms have greatly improved the performance of computer vision tasks. However, acquiring and annotating large-scale datasets for training such models can be expensive. In this work, we explore the potential of reducing dataset sizes by leveraging redundancies in video frames, specifically for instance segmentation. To accomplish this, we investigate two sampling strategies for extracting keyframes, uniform frame sampling with adjusted stride (UFS) and adaptive frame sampling (AFS), which employs visual (Optical flow, SSIM) or semantic (feature representations) dissimilarities measured by learning free methods. In addition, we show that a simple copy-paste augmentation can bridge the big mAP gap caused by frame reduction. We train and evaluate Mask R-CNN with the BDD100K MOTS dataset and verify the potential of reducing training data by extracting keyframes in the video. With only 20% of the data, we achieve similar performance to the full dataset mAP; with only 33% of the data, we surpass it. Lastly, based on our findings, we offer practical solutions for developing effective sampling methods and data annotation strategies for instance segmentation models. Supplementary on https://github.com/jihun-yoon/EVFR.

1. Introduction

Deep neural network architectures [15,27,28] and learning algorithms [20,33] have significantly improved the performance of computer vision tasks by leveraging large-scale datasets [10,22]. However, the cost associated with acquiring and annotating these datasets can be expensive. While much effort has been devoted to developing data-efficient methods for training neural networks, comparatively little research has focused on reducing dataset sizes by understanding the data itself.

One promising area of research is the exploration of semantic redundancies, which aims to reduce image classification datasets (e.g., CIFAR-10, CIFAR-100 [21], and ImageNet [25]) by discarding semantically redundant data. [29] concludes there is no redundancy for these datasets; however, [2] shows that 10% of the datasets can be reduced to achieve a performance of full dataset through agglomerative clustering [2] on feature representations. Another related area of study is video summarization, which aims to summarize the video content by selecting its representative video frames (keyframes), or video fragments (key fragments) [1,19]. [19] also utilizes a clustering on feature representations and picks frames close to the center of each cluster. However, [4] shows that K-means clustering on feature representations directly can result in cluster degeneracy, with one cluster dominating the others.

There are several studies on identifying or utilizing video keyframes. [7] utilizes video keyframes to improve inference speed and accuracy in video detection. The study defines a keyframe as a frame having small or fast-moving objects, making tracking results propagation difficult. [35] also utilizes keyframes for the same purpose with feature inconsistency between consecutive frames to define keyframes.

In this study, we explore the potential benefits of reducing redundancies in video frames in training datasets for instance segmentation. Our study does not aim to propose a novel adaptive sampling method superior to uniform sampling but to demonstrate the potential for reducing the cost of large-scale datasets while maintaining high performance and to provide insights on employing sampling methods through systematic experiments.

We hypothesize that representative frames with less redundancy can achieve comparable performance to dense datasets, demonstrating the potential for reducing the cost of data acquisition and annotation. To extract keyframes, we explore two sampling strategies that leverage the tempo-
The general nature of video frames. The first method, uniform frame sampling with adjusted stride (UFS), skips frames at a fixed interval. The second method, adaptive frame sampling (AFS), defines keyframes based on a frame variance index (FVI). We investigate two indexes defined by visual and semantic dissimilarity between consecutive frames. Our goal is to reduce the cost of generating training datasets; AFS utilizes simple learning-free keyframe extraction techniques, assuming no prior datasets. In addition, based on the keyframes, we demonstrate that a simple copy-paste data augmentation on the keyframes can bridge the big mAP gap caused by frame reduction.

Among the many available multiple object tracking segmentation (MOTS) and instance segmentation datasets, we choose the BDD100K MOTS 2020 dataset due to its large-scale, multiple classes, and objects per frame and long frame sequences with high FPS, which is best suited for our analysis although it still has some limitations. We employ the Mask R-CNN instance segmentation model and evaluate its mean average precision (mAP) on each down-sampled dataset.

Experiments reveal interesting findings that the performances of each sampling method vary across the size of the sampled dataset and show different performances on bounding box and mask prediction. As a result, we achieve similar performance to the full dataset mAP with only 20% of the data, and with only 33% of the data, we surpass it. Lastly, based on these findings, we offer practical solutions for developing effective sampling methods and efficient data annotation strategies for instance segmentation models.

The contributions of our work are summarized as follows:

- We show the potential of reducing the training dataset and its generation cost by using video keyframes.
- We show that higher frame rates do not always result in higher mean average precision (mAP) in instance segmentation tasks.
- We show that the performances of each sampling method vary across the size of the sampled dataset.
and show different effectiveness on bounding box and mask prediction.

- We show that a simple copy-paste augmentation on the keyframes can bridge the big mAP gap caused by reducing frames.
- Lastly, we offer practical solutions for developing effective sampling methods and efficient data annotation strategies for instance segmentation models.

2. Related works

2.1. Semantic redundancies and keyframe extraction

Semantic redundancy in image classification refers to the phenomenon where different images may contain similar semantic information or concepts, even if the visual features of the images are not identical. [2] conducted representative research to identify redundancy in image classification, arguing that 10% of images in the ImageNet dataset are semantically redundant. The proposed method uses agglomerative clustering [18] on images with cosine dissimilarities of feature representations obtained from pre-trained models. A similar approach has been applied to video summarization for keyframe extraction. K-means clustering [26] is applied on features extracted from frames, and keyframes are defined as centroids of each cluster. However, setting an appropriate K for a video can be challenging, and [4] show that K-means clustering after feature representation learning can result in cluster degeneracy, with one cluster dominating the others.

None of the previous research utilizes the keyframes to reduce training datasets. [7] proposes a adaptive selection scheme for keyframe extraction based on the observation that temporal propagation tends to be inferior to single-frame image-based detection when the objects are small and moving quickly. [35] also utilizes an adaptive keyframe selection method to improve speed and accuracy in video object detection and defines a keyframe using feature inconsistency between consecutive frames. However, two of the research have a limitation on the proper setting of the threshold to define keyframes, which can be tricky to be optimized.

2.2. Optical flow estimation

Horn and Schunck’s sparse optical flow [16] is one of the earliest methods for estimating optical flow, which uses a global energy minimization approach under the assumption of a smooth flow field. However, this method is limited by its sensitivity to noise and occlusions, leading to the development of dense optical flow methods such as Gunnar Farneback’s method [11]. Farneback’s method is based on the Lucas-Kanade algorithm, which tracks image gradients between two consecutive frames and introduces a polynomial flow field expansion to handle large motions and discontinuities. Deep learning-based methods, such as FlowNet [8], have shown great promise in optical flow estimation due to their ability to learn complex non-linear mappings, but still have limitations such as high computational cost and dependence on large amounts of training data. Recent works, such as LiteFlowNet [17], address these limitations using a lightweight architecture based on depthwise separable convolutions to reduce computational cost, and an unsupervised learning approach based on photometric loss to learn from unlabelled data.

2.3. Structure similarity index measure

The mean squared error (MSE) is a widely used image quality metric that measures the average squared difference between two images. However, the MSE does not consider the structural similarity between the images, which can be more important for visual perception. The structural similarity index measure (SSIM) [31] was developed to address this limitation as a more accurate image quality metric considering structural and pixel-wise differences between images. SSIM computes the similarity between two images based on luminance, contrast, and structure and has been shown to correlate better with human perception than MSE. As a result, SSIM has become a popular metric for evaluating the performance of image and video processing algorithms. However, despite its advantages, SSIM has some limitations, such as sensitivity to brightness and contrast changes and the need for careful parameter tuning.

2.4. Instance segmentation

Instance segmentation is a computer vision task that aims to detect and segment individual objects within an image. Over the years, numerous methods have been proposed to tackle this problem. Mask R-CNN [14] is a foundational two-stage object detection framework that extends Faster R-CNN [24] by adding a segmentation branch.

Cascade Mask R-CNN [3] extends Mask R-CNN by adding a cascade of Region Proposal Networks (RPNs) to improve object detection accuracy. It uses the output of each RPN to refine object proposals before passing them to the next RPN, resulting in a more accurate final set of proposals. Cascade Mask R-CNN also adds a separate mask branch for each cascade stage, allowing for more precise segmentation of objects.

Hybrid Task Cascade Mask R-CNN [5] further improves upon Cascade Mask R-CNN by adding a second task branch for predicting semantic segmentation. This allows the model to leverage both instance and semantic segmentation to improve the accuracy of object detection and segmentation. In this study, we employ Mask R-CNN, a foun-
dational instance segmentation framework to understand video frame redundancies in training data.

3. Method

Limitations of frame clustering on feature representations [4] and adaptive keyframe sampling with a threshold parameter [4, 26], we employ simple uniform frame sampling with adjusted stride and adaptive sampling with a weighted sampling based on a frame variance index defined by learning free visual and semantic dissimilarities.

3.1. Uniform sampling

The most commonly used approach for keyframe selection is uniform sampling. In this study, we employ a uniform frame sampling with adjusted stride (UFS). UFS is applied to each video clip $V$ at a fixed interval of size $s$ with a set of frames $\{f_1, f_2, ..., f_T\}$ with a length of $T$. With UFS, we select frames starting from the first frame $f_1$ and then skip frames at a fixed interval of size $s$. This results in a set of frames, denoted as $V[: s]$, which contains frames $\{f_1, f_1+s, ..., f_T\}$, where $k \in \{1, ..., n\}$ and $1 + s \times n < T$.

3.2. Adaptive sampling with a frame variance index

Although uniform sampling is simple and effective, it disregards that not all frames are equally important or influential in capturing scene variations. Therefore, a non-uniform frame selection strategy may be more desirable. We propose an adaptive selection scheme based on a frame variance index (FVI), which is calculated using the visual or semantic dissimilarity between consecutive frames. High FVI values indicate a frame is very dissimilar to the next frame, while low FVI values indicate the opposite. We calculate FVI for each frame with its next frame and for the last frame with itself. FVI is a normalized (scaling to unit length) weight representing frame redundancies and the sum of all the index results in 1. The figure 1 shows how FVI’s with SSIM, optical flow and cosine similarity of ImageNet pre-trained ResNet50 features change across video frames. To downsample a video clip $V$ into $n$ frames, we randomly sample $n$ frames using FVI as a weight for each frame. This weighted sampling ensures that not greedy but frames with lower variance are not overlooked. The sampling algorithm is presented in Listing 1. From now on, we introduce visual (Optical flow and SSIM) and semantic (ImageNet pre-trained ResNet50 [15] representations) dissimilarities for the FVI.

3.3. Visual and Semantic dissimilarities

Optical flow. Optical flow is the pattern of apparent motion of objects and surfaces between two consecutive frames caused by the movement of objects or cameras, represented by a 2D vector. While recent advancements in convolutional neural networks have led many researchers to introduce neural networks for optical flow estimation, we employ conventional methods to avoid the need for large-scale datasets.

$$I(x, t) = I(x + u, t + 1),$$

where $I(x, t)$ is the image intensity as a function of space $x = (x, y)^T$ at time $t$, and $u = (u_1, u_2)^T$ is the 2D velocity. The goal is to find a vector $u$, which is an optical flow vector. Under this assumption, Farneback’s dense optical flow estimation method [11] since accuracy is more critical than running time in our problem.

Optical flow estimation assumes that pixel intensities are translated from one frame to the next frame,
where \( N \) is the number of all pixels.

### 3.4. Structure similarity index measure

Structure similarity index measure (SSIM) is a method to measure the similarity between two images under the assumption that human visual perception is highly adapted for extracting structural information from a scene. Let's consider non-negative two image signal vectors, \( x \) and \( y \). SSIM is defined as following equation,

\[
\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{\mu_x^2 + \mu_y^2 + C_1}(\sigma_x^2 + \sigma_y^2 + C_2),
\]

where \( \mu \) is the mean intensity, \( \sigma \) is the standard deviation as an estimate of the signal contrast, and \( C_1 \) and \( C_2 \) are constants to avoid instability when \( \mu_x^2 + \mu_y^2 \) and \( \sigma_x^2 + \sigma_y^2 \) become zero. A higher SSIM value means higher similarity between two frames. We subtract SSIM value from 1 (which we call SSVD) to make a higher value means a higher difference like OFVD. A variance measure (visual dissimilarity) between two consecutive frames, \( f_t \) and \( f_{t+1} \), by SSVD can be defined as

\[
\text{SSVD}(f_t, f_{t+1}) = 1 - \text{SSIM}(f_t, f_{t+1}).
\]

### 3.5. Cosine similarity of feature representations

When given two frames \( f_1 \) and \( f_2 \), we extract their respective latent representations \( x_1 \) and \( x_2 \) using a pre-trained model. To measure their dissimilarity, we denote the dissimilarity between \( x_1 \) and \( x_2 \) using the cosine angle between them as follows [2] (we call this FSD):

\[
FSD(x_1, x_2) = 1 - \frac{(x_1, x_2)}{\|x_1\| \|x_2\|}.
\]

While latent representations can be obtained from any neural network, we use the ImageNet pre-trained ResNet50 model as it is the most representative initial feature extractor and does not require additional training. Additionally, we select the last average pooling layer among several choices of the network’s layers as it can identify the largest redundancy [2].

Figure 3. The number of samples of each dataset and sample ratio are shown in this figure. The y-axis on the left indicates the number of samples, and the y-axis on the right indicates the sample ratio(%). S1 is a full dataset consisting of 154 video clips, each consisting of roughly 200 frames (a few have around 100 frames).

### 4. Experiments

#### 4.1. BDD100K MOTS dataset

The BDD100K dataset [34] is a large-scale dataset designed for autonomous driving applications, containing 100K diverse video clips and several datasets for different tasks. However, the instance segmentation dataset...
(BDD100K IS)\(^1\), consisting of 10,000 video clips, only includes a few labeled frames every 10 seconds of each video recorded for approximately 40 seconds, resulting in only about four frames. Moreover, there is no information available to associate each series of frames to each video in the file names. Because of this limitation, we use the Multiple Object Tracking and Segmentation (MOTS) dataset, which has more dense annotation (recorded at 30 FPS and labeled at 5 FPS) and is larger in scale than other MOTS datasets, as shown in Tab. 1. YouTube VOS [32] has more frames than BDD100K MOTS, but the annotation per frame is only 1.64, almost a single object. As our goal is to detect and segment as many objects as possible while considering difficulty, BDD100K MOTS is the best choice (however, 5 FPS is still not enough to see trade-offs between accuracy and sampling). We use original train data for training and validation and validation data for testing. Training and validation data include 154 video clips resulting in about 30,745 frames, and test data include 32 video clips, resulting in 6,475 frames. The dataset consists of seven classes: pedestrian, rider, car, truck, bus, motorcycle, and bicycle.

Figure 4. Log values of each class count. The distributions shows our adaptive sampling method does not distort the original class distribution while selecting frames.

### 4.2. Down-sampled datasets

We utilize the full dataset to establish an ideal performance reference. Then, we apply uniform sampling to the dataset by adjusting the stride (UFS) to 2, 3, 4, 5, 7, and 10. Figure 3 illustrates the number of samples for each down-sampled dataset. As indicated in figure 3, we follow a naming convention for each sampling ratio to facilitate easy reference. Subsequently, we apply our adaptive sampling method (AFS) and random sampling (RFS) to the full dataset, each with the same number of samples as the uniformly down-sampled datasets. For AFS and RFS, all frames can be selected once without replacement while maintaining the class distribution. Figure 4 shows the logarithmic values of each class count across the datasets, which exhibit a consistent trend.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>KITTI MOTS [12]</td>
<td>8K</td>
<td>21</td>
<td>7.49</td>
<td>38K</td>
</tr>
<tr>
<td>MOTS Challenge [30]</td>
<td>2.9K</td>
<td>4</td>
<td>228</td>
<td>27K</td>
</tr>
<tr>
<td>DAVIS 2017 [23]</td>
<td>5.2K</td>
<td>90</td>
<td>197</td>
<td>-</td>
</tr>
<tr>
<td>YouTube VOS [32]</td>
<td>120K</td>
<td>4.5K</td>
<td>7.8K</td>
<td>197K</td>
</tr>
<tr>
<td>BDD100K MOTS [34]</td>
<td>14K</td>
<td>70</td>
<td>6.3K</td>
<td>129K</td>
</tr>
</tbody>
</table>

Table 1. Comparisons with other MOTS and VOS datasets. The table is referenced from [34].

### 4.3. Experiments settings

To investigate the impact of frame redundancies on instance segmentation model mAP, we conducted experiments using the widely-used Mask R-CNN model [14] with ImageNet pre-trained ResNet50 backbone [15] and feature pyramid network (FPN) in MMDetection library [6]. We used synchronous batch normalization with a batch size of 16 and the SGD optimizer.

To ensure fair comparison across datasets, we trained the models for almost the same number of iterations, considering the size of each dataset. Specifically, if the model learned \( M \) samples of the full dataset for \( N \) iterations, we trained the model using \( M/2 \) samples of the dataset for \( 2N \) iterations. This approach allowed us to compare the effect of each data value on the model’s performance without the confounding factor of learning more data points. However, to avoid learning exactly the same data, we use the standard scale jittering [13] for basic data augmentation, which is resizing and random cropping.

We trained models with the full dataset for 13 epochs, and, as explained earlier, down-sampled datasets were trained for almost the same number of iterations with the full dataset. All models’ mAP started to converge around iterations of 9 epochs in the full dataset\(^2\). We conducted each experiment with different random seeds thrice and averaged the results since many experiments depend on randomness (e.g., sampling, augmentation). We evaluated models by the best performance during training, and employed the COCO [22] mean average precision (mAP) for model evaluation metrics, averaging mAP across different intersections over union (IoU) from 0.5 to 0.95 by every 0.05.

### 4.4. Keyframes in training data

Firstly, we demonstrate the existence of keyframes in training data by comparing uniform sampling (UFS) with random sampling (RFS). As shown in the figure 5 and 6, all UFS results surpass the RFS results. Randomly selected

---

\(^1\)To ensure the BDD100K instance segmentation dataset is different from MOTS, we refer to it as BDD100K IS. However, it is not an official name.

\(^2\)More experiment details on https://github.com/jihun-yoon/EVFR
frames may not be well distributed in time, and the selected frames are not representative enough to cover whole video frames. In addition, UFS S2 and S3 datasets achieve higher mAP than the full dataset S1 for both BBox and Mask mAP. This suggests that redundant frames in S1 may degrade the mAP, increasing the generalization error. Although a higher FPS generally leads to higher mAP, our results provide a counterexample. Notably, the mAPs between UFS S4 and UFS S5 are almost similar, while RFS S4 and RFS S5 have more significant differences for BBox mAP. This indicates that the additive frames from UFS S5 to UFS S4 may not have more information to learn and do not lead to a performance increase.

The results show that AFS outperforms RFS in most cases. This indicates that AFS successfully selects better keyframes. Furthermore, in the S4 setting, AFS outperforms Uniform Frame Sampling (UFS) or performs similarly. When we look at differences between FVI methods, AFS with SSVD performs well overall, but AFS with FSD only outperforms UFS Mask mean average precision (mAP) in the S4 setting.

The results show that UFS is simple and very effective overall. However, we assume if the full dataset has a higher FPS, there is much possibility that AFS works better. Because the full dataset (S1) result shows that frames are more redundant than S2. And UFS S2 or S3 sampling process skips only one or two very redundant frames uniformly, ideally. This is a limitation of the current dataset. We need another dataset to verify this for the future research.

However, AFS fails when the number of samples is too small. Since the number of full frames for each video clip is approximately 200, the number of frames for each video clip is approximately 20 in the S10 setting. Adaptive sampling methods may fail to cover diverse frames across the time axis. We designed adaptive sampling as a weighted sampling (AFS) not to become greedy on higher weights, but it is still insufficient to spread over the time axis (S10 results in the figure 2). Therefore, from a different viewpoint, it is better to employ uniform sampling when the number of samples is small.

4.6. Bridging performance gap with simple copy-paste

Lastly, we demonstrate that a simple copy-paste augmentation [13] on keyframes can bridge the big mAP gap caused by reducing frames. When video is not long enough or dynamic, we expect the copy-paste data augmentation on keyframes can make enough diversity as much as a dense dataset or more. To verify this, we employ a simple copy paste (SCP) data augmentation. The figure 7 shows that, with just 20% of data (S5), we achieve similar performance.
Figure 8. Normalized standard deviations of OFVD for each video are shown in this figure. The scale of deviation means the dynamicity of the video. A high value means high dynamicity. Videos are sorted in increasing order, and the video with the highest value is a video (name:00207869-902288d1), which is shown in the figure 1. The video name reference for each deviation is on https://github.com/jihun-yoon/EVFR.

as the full dataset mAP, and with only 33% of data (S3), we surpass it.

Figure 9. Frame variance index (FVI) measured by OFVD of an example video (name:0024b742-acbed4fb) is shown in the figure. This video has the lowest deviation, located at the most left in the figure 8. The plot is more static than the figure 1, which has the highest deviation.

4.7. Discussion

Based on our findings, we suggest a data annotation strategy to build the initial dataset. Firstly, start with a small number of annotations uniformly. Secondly, quantify the dynamicity of the video with the frame variance index and choose different sampling methods by the dynamicity. Lastly, we can leverage the performance highly with a simple copy-paste augmentation.

However, certain limitations need to be addressed in future work. Firstly, We can see that each dataset has a different variance scale, and video clip with high deviation has more dynamic frame changes in the figure 8. Since UFS works well for static video, we can use different sampling methods on the dynamicity of video.

Secondly, our current weighted random sampling method cannot cover diverse frames across time when the number of samples is too small. Incorporating this consideration in the design of adaptive sampling could significantly enhance performance.

Thirdly, we observed that semantic dissimilarity exhibits a high potential for sampling frames useful for both Bounding Box (BBox) and Mask prediction in figure 5 and 6. We expect that more distinct feature representations can be developed for keyframe extraction with the help of strong self-supervised methods [9]. This is another promising direction for future research.

5. Conclusion

In this study, we investigated the impact of frame redundancies on instance segmentation model accuracy using the Mask R-CNN model. Through our experiments, we demonstrated the existence of keyframes in training data and the different effects of UFS and AFS for keyframe extraction in the video. UFS experiments showed that higher FPS does not always lead to higher performance. AFS outperforms RFS in most cases and UFS in the S4 setting but fails when the number of samples is too small. Additionally, we showed that a simple copy-paste augmentation on keyframes could bridge the big mAP gap caused by reducing frames. Overall, our findings provide insights into the impact of frame redundancies on model accuracy and offer practical solutions for developing effective sampling methods and data annotation strategies for instance segmentation models.
References


