# SEMI-AUTOMATED LABELING OF VIDEO USING ACTIVE LEARNING FOR OBJECT DETECTION

By

Roberto Muntaner Whitley

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical and Computer Engineering – Master of Science

2023

## ABSTRACT

Labeling video sequences is a critical task that is required for a wide range of supervised learning applications. In general, manually labeling videos is an extremely repetitive and time-consuming task. Often, the process is sped up by sharing the workload across multiple workers, but this can create other problems, such as varying quality and consistency of labels. Meanwhile, the area of active learning has been proposed for assisting in the labeling of images for classification and object detection tasks. However, minimal prior work is centered around the utility of active learning for video labeling. In this thesis, we attempt to address the gap in prior efforts by proposing a Semi-Automated Labeling of Video (SALV) framework using active learning to support supervised object detection applications. Firstly, we propose a general architecture for the SALV framework that is built on intra-video training and testing. The proposed SALV architecture exploits the fact that labeling video provides a unique opportunity where training and testing can be performed on consecutive frames that contain highly correlated information. Secondly, we incorporate traditional active learning methods that utilize the confidence values produced by detections to select important frames for the next iteration. Thirdly, we propose two strategies for active learning of video labeling: minimal-Distance Iterative Active Learning (min-DIAL) and maximal-Distance Iterative Active Learning (max-DIAL). Lastly, we explore information theory to select frames with the most diversity using the Jensen-Shannon divergence to calculate the difference between certain frames based on the location of detections. We analyze the performance of the proposed SALV architecture in terms of the time taken to complete the labeling of the video sequences and present our results using the popular KITTI Tracking dataset. We show that our proposed max-DIAL framework is the most efficient method and can reduce the time taken to label video by a factor of 10.

## ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Radha, for his belief, patience, and guidance. Without his belief, I would not have had the opportunity to join the Connected and Autonomous Networked Vehicles for Active Safety (CANVAS) group and conduct research in a field that I find intriguing. His patience allowed me to explore unfamiliar areas and generate the knowledge needed to complete my research without any added pressure. His knowledge and guidance were pivotal in helping me achieve my goals, which I will always remember and be extremely grateful for. Without him, I would not be where I am today.

I would also like to thank my committee members, Dr. Morris and Dr. Bopardikar, for their willingness to assist in the completion of my thesis. I value your time, knowledge, and opinions, and appreciate your flexibility during these busy times.

## TABLE OF CONTENTS

# INTRODUCTION

The vast amount of data available has allowed the area of deep learning to advance dramatically over the past decade. In particular, an increase in accessible labeled data has enabled a wide range of supervised deep learning models to become state-of-the-art for many applications and emerging services. In general, increasing the amount of data used during training improves the performance of the model, considering that the data is relevant and diverse. However, in a supervised learning environment, ground truth labels are required to train the models. Despite recent advancements in developing a variety of software annotation and labeling tools, ground truth labels are generally produced manually by humans. For tasks like object detection, where each object requires its own bounding box, this process is extremely time-consuming.

ImageNet [1] was created at a time when most researchers were heavily focused on designing new machine learning models when the more pressing issue was the lack of large-scale datasets to train current models. However, after early calculations highlighted an unrealistic timeline to create the enormous dataset, crowdsourcing was explored instead. Amazon Mechanical Turk (AMT) [2], the crowdsourcing marketplace used for creating ImageNet, is a platform that allows businesses the opportunity to utilize endless remote workers for a wide range of demanding tasks. In over 2 years, using more than 25,000 AMT workers [3], ImageNet was created. However, while the time to create the dataset was reduced, the cost of creating the dataset was increased. ImageNet was only possible with support from several sponsors; therefore, some researchers and smaller businesses may not have sufficient funding to explore crowdsourcing options.

Other factors that need to be accounted for when planning to manually label data are the size and experience of the group of annotators. The larger the group, the larger the variation in the quality of the ground truth labels. The smaller the group, the larger the workload for each individual. This is likely to result in error-prone work due to tedious, repetitive actions. A lack of understanding in advanced areas, which is likely to be extremely common on crowdsourcing websites, can further decrease the quality and consistency of the labels.

In this thesis, we propose a Semi-Automated Labeling of Video (SALV) framework that minimizes the amount of human interaction required during the labeling process. Combining a small subset of manually labeled images with current deep learning applications, the amount of human input can be greatly reduced, while simultaneously improving the accuracy and consistency of the labels. Leveraging the fact that consecutive frames in a video sequence share many similarities, we manually label a small subset of the data that are used to train an object detector. The trained object detector is used to predict objects in the remaining frames of the video. As many of the objects used to train the model will also appear in the unlabeled frames, perhaps closer or at a slightly different angle, the detections are extremely accurate. After testing the object detector on the remaining unlabeled frames, human verification is required to fix any false positives (FPs) or false negatives (FNs). The highly accurate detections greatly reduce the time required for human verifiers to add or remove any bounding boxes. The main contributions of this work include:

- A Semi-Automated Labeling of Video (SALV) architecture that exploits an intra-video training and testing strategy. The proposed SALV framework is built on the notion that video labeling provides a unique opportunity where training and testing can be accomplished over the same dataset with highly correlated information.

- Minimal-Distance Iterative Active Learning (min-DIAL) and maximal-Distance Iterative Active Learning (max-DIAL) strategies for the SALV framework. They are simple but effective approaches for iteratively selecting new, unlabeled video frames based on their respective locations in the video sequence.
- A Jensen-Shannon Diverge (JSD) metric used to calculate the distance between frames based on their distributions. Each frame is divided into 8 sections and the distribution is based on the number of detections that are present in each section.
- We analyze the performance of the proposed SALV architecture based on the time taken to label video sequences using traditional active learning methods, our proposed min-DIAL and max-DIAL approaches, and the JSD metric. Our analysis is presented using the popular KITTI Tracking dataset [17].

# CHAPTER 1: CURRENT LITERATURE

It is well understood that supervised deep learning models require an enormous amount of labeled data for training. However, many researchers are still focusing their attention on building more accurate models using current datasets, rather than producing more efficient ways to label new data. Nevertheless, different directions have been explored to combat the bottleneck caused by high annotation costs.

## *1.1 Linear Interpolation*

Utilizing the spatial-temporal characteristics of a video, [4] estimates object locations between manually labeled keyframes using linear interpolation and homography-preserving techniques. Although this technique could be beneficial for basic videos, for applications like autonomous driving, where vehicles are accelerating and decelerating at random and unpredictable rates, the majority of the predicted bounding boxes are likely to be incorrect.

## *1.2 Weakly Supervised*

More recent work has focused on weakly supervised labeling techniques. Instead of drawing a bounding box around an object, researchers have attempted to create ways to label objects in a much less time-consuming manner. [5] proposes a center-click technique, requiring the human annotator to click where they imagine the center of the bounding box around the object would lie, reducing labeling time by more than 9x. [6] uses class labels to indicate what object categories belong to the image, without any type of localization of the objects present, due to the vast number of image-level annotations available on the internet. However, while these techniques greatly reduce the annotation time, the accuracy of the model is also significantly reduced. [7] only requires human annotators to verify bounding boxes produced automatically during an iterative learning process, reducing annotation time by more than a factor of 6x while

performing better than weak supervision. Despite these improvements, fully supervised learning remains the most accurate method for training object detectors.

### *1.3 Efficient Labeling*

The traditional way of drawing bounding boxes, where annotators click and drag a box to enclose an object, is often cognitively demanding and inefficient. Therefore, [8] proposes a more natural way for human annotators to label objects. Instead of drawing a box around the object of interest, annotators are asked to click the 4 extreme points of the object: the top, bottom, left, and rightmost points. This simple but effective difference allows annotators to label boxes 5x faster than regular bounding box annotations while maintaining the same quality.

### *1.4 Object Detection and Tracking*

Although objects may vary in different environments, many features can be learned and transferred to different situations. [9,10] both use a pre-trained object detector and tracker to label all frames in a video. Afterward, the predictions are passed to a human for verification and correction. Although both papers reduce the time required to label their respective datasets, this technique is only possible if similar datasets are available for pretraining.

The most similar work to ours is [11] which uses a manually labeled subset to train a model before testing on all remaining frames. However, the dataset used in this paper is a relatively simple indoor dataset with the majority of the objects being fire extinguishers and chairs that were recorded on a hand-held device. Unlike our work, they use a single iteration which restricts the learning of the model as it sees new data.

### *1.5 Active Learning*

Active learning (AL) has become a widely used technique for selecting specific samples from a dataset. As different objects provide different amounts of information to the model,

selecting images that contain the most information has proven to be a more efficient method for data labeling. Although the majority of active learning research is based on image classification, recent work has extended this method to the more challenging task of object detection [12,13,14]. The general idea of active learning is to randomly choose a subset of unlabeled data. A human annotator manually labels these samples and uses them to train an object detector, which is then tested on the remaining unlabeled samples. The detections produced are used to formulate an uncertainty measure for each image that can be used to extract the images that the model struggles with the most. These often correspond to the objects that provide the most information to the model, considering they are the most challenging. As many of the images are likely to contain multiple object instances, there are many ways to derive an uncertainty value for a specific image. [12] experiments with the confidence values produced by the bounding box detections, a natural extension to the techniques used in image classification. Summing, averaging, and taking the minimum confidence value are just a few different ways that confidence scores can be used to measure the uncertainty of an image. However, as the images contain more objects from a range of different classes, these techniques become less effective. [13] and [14] propose more advanced approaches that use adversarial instance classifiers and mixture density networks, respectively, to learn the uncertainty of different images. [15] further extends active learning for object detection into video, using temporal coherence to detect where FPs and FNs may have occurred.

# CHAPTER 2: SPATIAL-TEMPORAL COHERENCE

## *2.1 Intra Training and Testing*

Modern cameras are capable of capturing an extremely large number of frames every second, providing a smooth visual experience. However, this produces a high level of similarity and redundancy when manually labeling object instances in consecutive frames. Even when a vehicle is driving at a relatively high speed, the surrounding scene barely changes on a frame-by-frame basis.

In traditional object detection models, the images used during training are required to be different from the images used for testing. This allows the performance of models to be compared fairly when tested on images it has never seen before. However, for applications such as automated labeling, we can approach the problem from a different angle. If, for example, we looked at the 1st and 5th frames of a video sequence, it is highly likely that the majority of the objects in the 1st frame are also present in the 5th frame. Of course, these objects could move closer, further away, or become partially occluded, but the same objects are often present for several consecutive frames. Therefore, as an example, if we trained an object detector using the 1st and 5th frames of a video sequence, we would expect it to perform well when tested on the $2^{nd}$, $3^{rd}$, and $4^{th}$ frames. A visual example of this methodology can be seen in Figure 1.

Using this logic, we create initial subsets using a range of different subsampling rates. These initial subsets are used to train an object detector which is then tested on the remaining frames. We decided to use YOLOv5 [16] for this task because of its speed, accuracy, and usability. It is important to note that 100% of the dataset is used each time. If a sampling rate of 10 is used for training, then 1 in every 10 frames (including the first frame) is uniformly added to the training set, with the other 9 being added to the test set. This means that for different

sampling rates, the train and test sets will be different sizes. Although this is normally bad practice when comparing models, for this application we are more concerned about minimizing the size of the train set while maintaining a high enough accuracy on the test set. For fairer testing, the higher the sampling frequency, the more epochs were allowed during training to offset the smaller number of training samples.

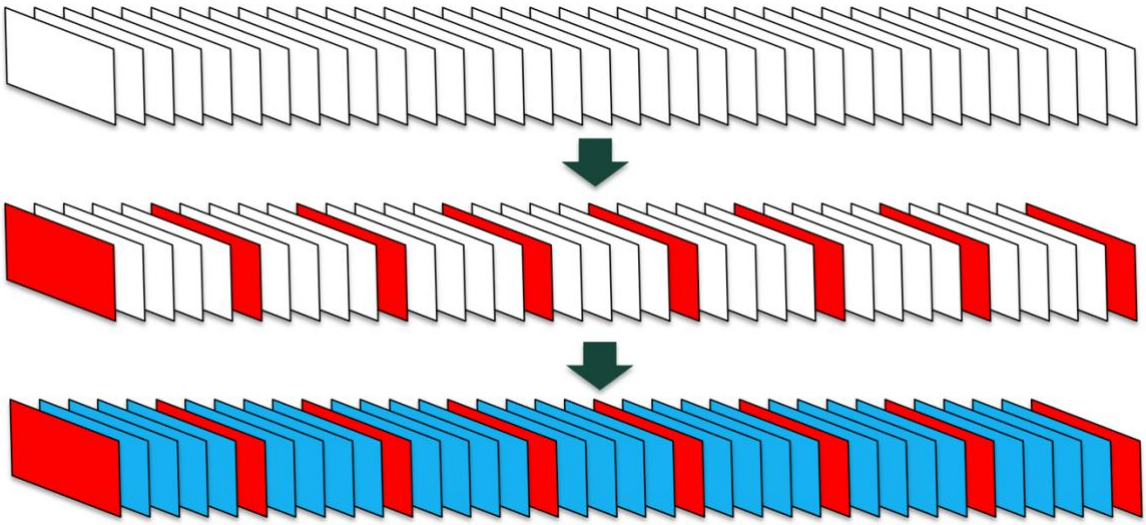

*Figure 1. Initially, all frames are unlabeled (white). We manually label every 5th frame (red) which are used to train our model. The model is then tested, verified, and corrected on all remaining frames (blue).*

*2.2 Dataset*

Although the primary target of our proposed framework is an unlabeled dataset, to show the effectiveness of our method, we must exploit a publicly available dataset that provides ground truth annotations. Although there are many suitable datasets accepted by the broader research community, this thesis focuses on the KITTI Tracking [17] dataset due to its small size and ease of use. It is important to highlight that we used this dataset because it is a collection of *video* sequences and not a set of isolated images. Moreover, although the KITTI dataset is a well-

established benchmark, some bounding boxes need to be added or removed to provide accurate results for our application. These consist of:

- Objects from preceding frames that are still labeled although no longer visible.
- Objects that have become fully occluded but are still labeled although no longer visible.
- Some objects of interest that are not labeled.

As many of the ’Van’, ’Car’, and ’Truck’ classes are very similar, and labeled inconsistently, we combined them into a single ’Vehicle’ class. The ’Pedestrian’ class is the only other class that has a reasonable number of instances for consideration. However, many of the labels contain only part of a pedestrian or multiple pedestrians within the same bounding box. The ’Tram’, ’Person Sitting’, ’Misc’, and ’Cyclist’ classes were also removed due to insufficient object instances; therefore, the ’Vehicle’ class is the only class considered in this thesis.

*2.3 Evaluation Metric*

Considering that our goal is to minimize the amount of human interaction required to label the dataset, we need to calculate an accurate estimation for the time taken to complete each experiment. The initial labeling consists of drawing bounding boxes from scratch around all the object instances in the uniformly subsampled frames. After training our model and then testing on a specific subset, a human is required to verify each frame to remove any FPs and add any FNs. After testing our model on the frames of interest, we are provided with values for the precision and recall for that subset. This allows us to calculate the number of false positives and false negatives using Equation 1 and Equation 2, respectively. Our experiments use 6255 frames containing 29,080 vehicle instances, which the train and test split will always sum to. We used an intersection-over-union (IOU) threshold of 0.6 to calculate the precision and recall values because many of the KITTI labels are inconsistent and insufficiently tight around many of the

object instances, as seen in Figure 2. This leads to many extremely accurate detections, and even tighter around the object than the ground truth label, being counted as false positives at higher IOU thresholds.


*Figure 2. The bounding boxes for many object instances in the KITTI dataset should be tighter, which would allow more accurate IOU calculations.*

$$False\ Positives = Instances * (1 - Precision) \quad (1)$$

$$False\ Negatives = Instances * (1 - Recall) \quad (2)$$

We use the information provided in [18] as a good estimate for the time taken to complete the different tasks. Although they provide both median and mean values for each task, we opted to use the median values. This is because they show that there are a minority of workers that take an unreasonable amount of time, which may be caused by taking breaks or not performing the tasks properly. The median time for drawing a single bounding box is 34.5s, which includes a 9s quality verification check. They also provide a 'Coverage Verification' time of 7.8s for measuring how long it took annotators to scan an image for all instances. This time will be used for the task of scanning each frame to find any FPs and FNs produced by our detector. We use a modest value of 8s to remove FPs, based on the fact it is a much simpler task than verifying the quality of the box. Therefore, the total time consists of manually labeling the

initial bounding boxes to train our model, scanning the frames that our model was tested on, removing any false positives, and finally drawing any false negatives that were missed, shown in Equation 3. Note that we have not included the time taken to train the model as we are more interested in minimizing the amount of human interaction.

$$Total\ Time = Initial\ Label + Scan + Add\ FN + Remove\ FP \quad (3)$$

*2.4 Labeling Times*

We calculate the time taken to label the full dataset by training our model using different subsampling frequencies. The model is then tested on all remaining frames before verifying and correcting detections. We sum the time it would take to draw initial boxes, scan test frames, and add/remove any boxes, as shown in Table 1. When manually labeling the whole dataset (frequency of 1), we are not verifying or removing any boxes. For any of the splits afterward, the more frames that we manually label for training, the less we have to verify and correct, and vice versa. While varying the subsampling rate alters the number of frames that need to be manually labeled and verified, it also affects the accuracy of the model. Generally, the more frames used for training, the more accurate the model will be. However, as manual labeling is extremely expensive and time-consuming, we are looking for an optimal subsampling rate that uses the fewest number of frames to provide a reasonably high accuracy. The total times from Table 1 can be visualized easier in Figure 3, where the optimal subsampling frequency is 20. Using a subsampling rate of 20 takes an estimated time of 44.99 hours, which is around 16% of the estimated time taken to manually label the whole dataset.

| Frequency | Initial | Scan | Add | Remove | Total (hours) |
|---|---|---|---|---|---|
| 1 | 278.68 | 0 | 0 | 0 | 278.68 |
| 2 | 139.41 | 6.78 | 1.95 | 0.65 | 148.79 |
| 5 | 55.72 | 10.84 | 5.58 | 1.19 | 73.33 |
| 10 | 28.09 | 12.2 | 8.52 | 2.56 | 51.37 |
| **20** | **14.26** | **12.87** | **14.55** | **3.31** | **44.99** |
| 30 | 9.11 | 13.1 | 18.59 | 4.37 | 45.17 |
| 40 | 7.08 | 13.21 | 22 | 4.41 | 46.7 |
| 50 | 5.76 | 13.28 | 22.38 | 5 | 46.42 |
| 60 | 4.61 | 13.33 | 30.7 | 5.02 | 53.66 |
| 70 | 4.07 | 13.36 | 25.82 | 5.67 | 48.92 |
| 80 | 3.51 | 13.38 | 31.1 | 5.17 | 53.16 |
| 90 | 2.88 | 13.4 | 36.13 | 5.82 | 58.23 |
| 100 | 2.94 | 13.42 | 31.16 | 5.63 | 53.15 |
| 160 | 1.61 | 13.47 | 39.62 | 6.04 | 60.74 |
| 320 | 0.72 | 13.51 | 40.31 | 7.41 | 61.95 |
| 640 | 0.31 | 13.53 | 54.84 | 6.39 | 75.07 |
| 1280 | 0.03 | 13.54 | 98.09 | 6.2 | 117.86 |

*Table 1. Time taken to label the full dataset. Different subsampling rates vary the number of frames that need to be manually labeled for training the model.*

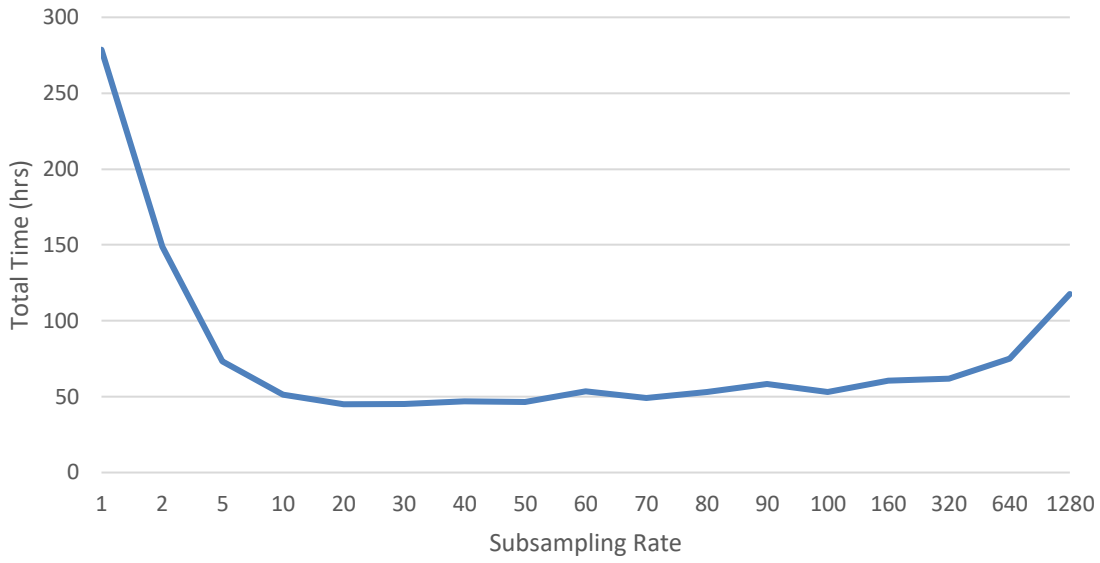


*Figure 3. Total time to label, verify, and correct using different subsampling frequencies.*

# CHAPTER 3: ACTIVE LEARNING

Although recent work using active learning has produced more complex algorithms for selecting uncertain images, due to the highly accurate detections produced by the initial stage of our model, we explore low-complexity active learning strategies to further improve our framework. After manually labeling the subsampled frames (anchor frames), we train our model and then test on all remaining frames. However, unlike our previous method, we use the confidence values from the outputs to select the more difficult frames. We know that the detections from our model are extremely accurate, allowing us to locate the frames that the model struggles with the most. These frames likely provide our model with the most information and allow us to perform an iterative process that updates our model as it sees more data. After selecting the data to use for the next iteration, we verify, correct, and add those samples to our training set. We train our model with our updated train set before testing on all remaining frames, with an example for uniformly sampling every 10 frames shown in Figure 4. This iterative active learning process can be repeated multiple times, but it is important to note that each iteration requires retraining of the model.

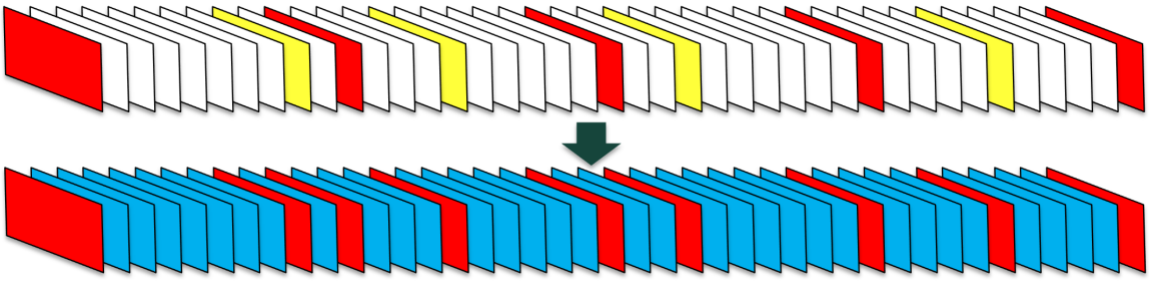

*Figure 4. The anchor frames (red) are manually labeled and used to train our model. All frames between the anchor frames are tested and the most informative frames (yellow) are selected. The selected frames are verified, corrected, and added to the train set. The model is retrained then all remaining frames (blue) are tested, verified, and corrected.*

A high-level architecture for our proposed Semi-Automated Labeling of Video (SALV) framework can be seen in Figure 5, where the goal is to minimize the amount of human interaction required to label visual data. Starting with a completely unlabeled dataset, an initial subset is selected to be manually annotated. These annotations provide the building blocks for our model to label selected frames from our iterative active learning process.

### *3.1 Selecting Uncertain Frames*

There are many techniques used to select the most uncertain frames from a dataset. However, we use a simple, more traditional method that utilizes the confidence values of the detections provided by our model. Along with bounding box predictions, object detectors provide confidence scores that express how confident the model is that the bounding box encloses the correct object. In general, high confidence values are given to detections that are relatively simple objects that the model has no trouble identifying. Whereas lower confidence values are often given to objects that are smaller, further away, or partially occluded. These more challenging objects are what we are interested in as they provide the model with more information than objects it already detects with ease.

With the majority of frames containing multiple object instances, there are many ways to calculate an overall confidence score for each frame. Averaging the individual confidence scores within a frame allows there to be no bias regarding the number of detected objects in that particular frame. Calculating the median value for each frame allows outliers to be overlooked and is more likely to select frames with multiple low-confidence detections. Using the lowest confidence value from each frame to represent the overall confidence value allows us to select frames with a challenging object in it but doesn’t consider any of the other objects that are

present in that frame. Examples of detections and their confidence values can be seen in Figure 6 and Figure 7.

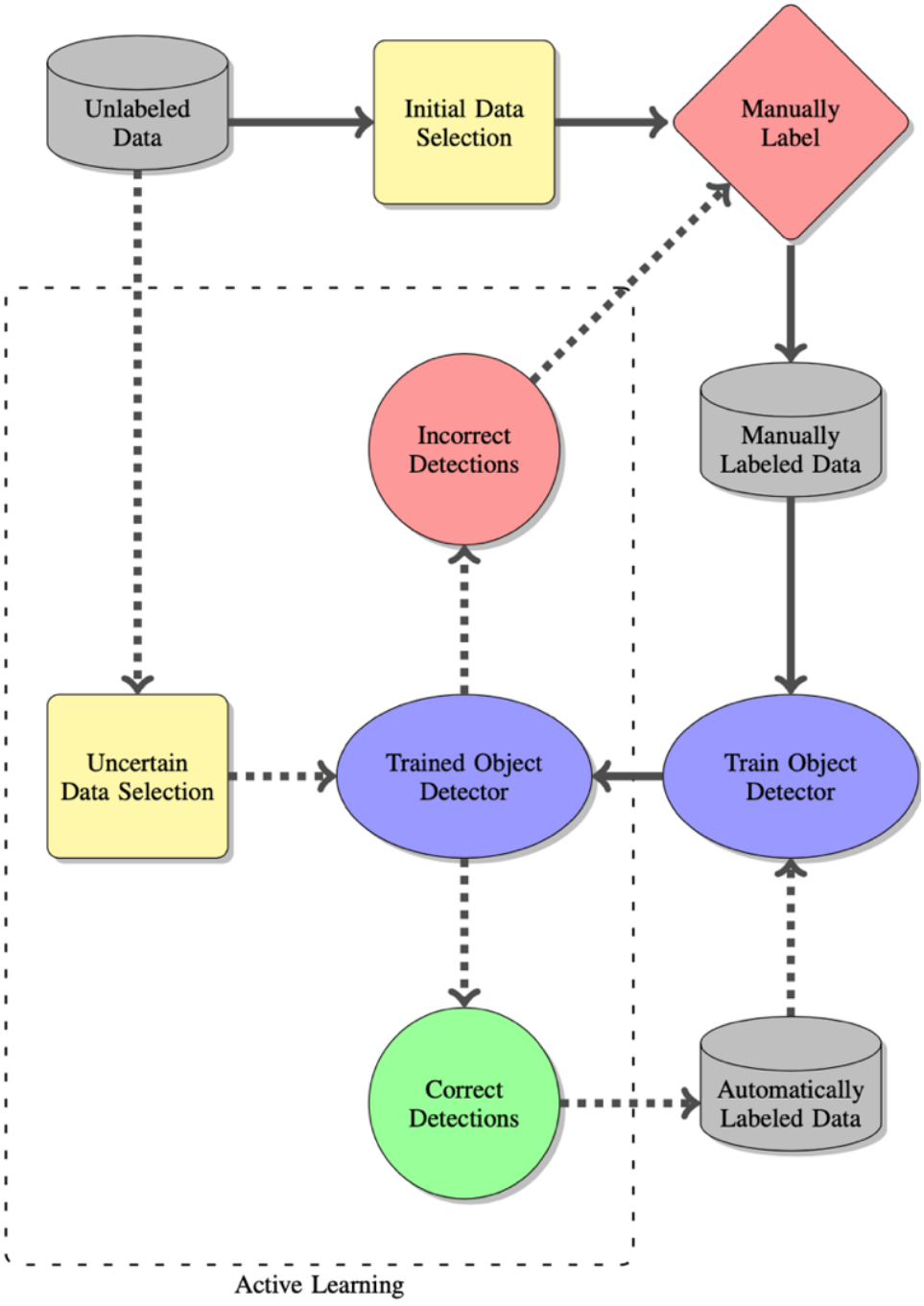


*Figure 5. Architecture of our SALV framework. After initially training our model, uncertain data is selected based on several different metrics.*


*Figure 6. Frame containing 3 detections with 0.99 confidence. The average, median, and lowest values are all 0.99, so it's unlikely this particular frame will be selected in any of our active learning algorithms.*


*Figure 7. Frame containing 5 detections with varying confidence values. The average is 0.742, the median is 0.97, and the lowest value is 0.34. This frame is likely to be selected when using the average or single lowest value as an uncertainty measure but unlike.*

### *3.2 Comparing Uncertainty Selections*

To understand the distribution of which frames have the lowest uncertainty at each iteration, it is important to generate histograms that help visualize any patterns during the selection process. A measurement is given to selected frames based on their distance from the anchor frame to its left. An example of the distance calculation can be seen in Figure 8, where the yellow frames would be selected based on their uncertainty value. After selecting the most uncertain frames and calculating their respective distances, we can sum up the number of occurrences for each distance. The examples shown in Figure 9 are for a model that was trained

on every 20th frame, meaning 19 possible frames can be selected between each anchor frame. Apart from the average, median, and single lowest confidence values, we also randomly selected frames between each anchor frame for comparison. It is interesting to note that for all of the average, median, and single lowest confidence values, more frames are being selected from around the middle. This is expected because the anchor frames are used to train the model, so the majority of the frames close to the anchor frames are the most similar and contain many of the same object instances.

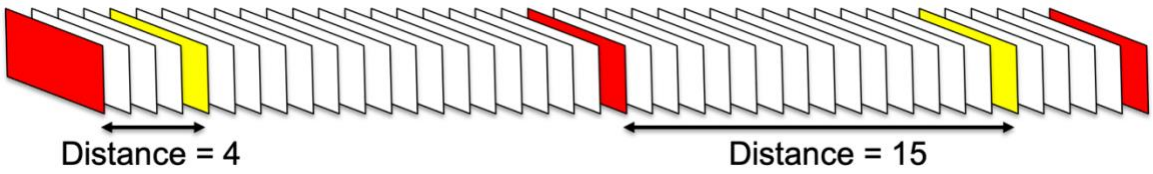


*Figure 8. Each selected frame's (yellow) distance is measured from the anchor frame (red) to its left. The anchor frames are used to train the model, and the selected frames are chosen based on their average, median, or single lowest confidence values.*

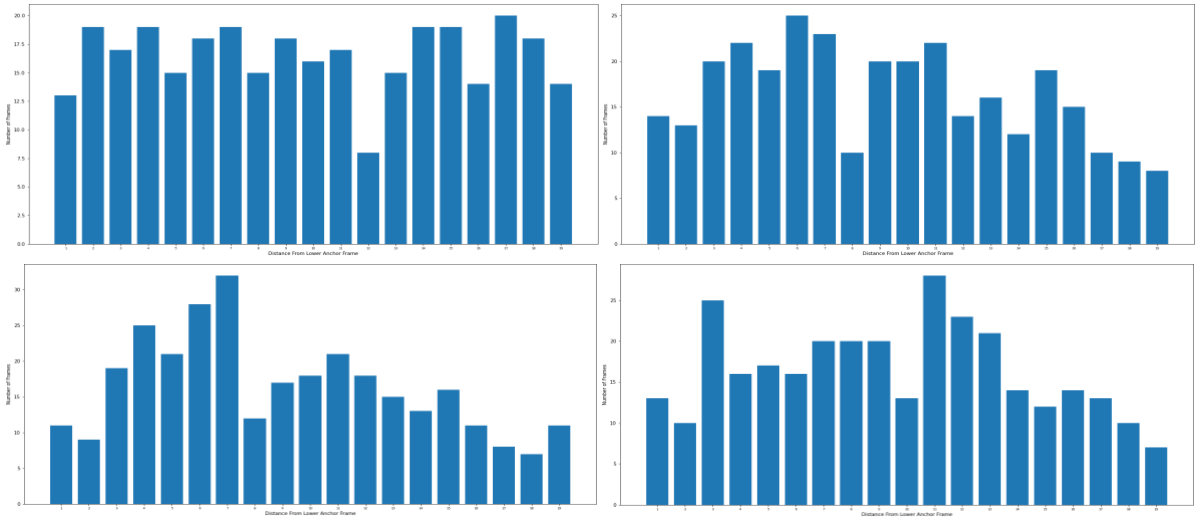

*Figure 9. Top Left: Uniformly randomly selected frames show the flattest distribution, as expected. Top Right: Lowest single confidence value, Bottom Left: Average confidence value, Bottom Right: Median confidence value.*

*3.3 Comparing Times*

When comparing times, it is important to note that the higher the initial subsampling rate, the more active learning iterations can be performed. When initially training the model with a subsampling rate of 10, we only perform one active learning iteration. This is based on the fact that the detections at this point are extremely accurate and performing another iteration barely improves the model. However, when we initialize with a subsampling rate of 80, we perform 4 iterations. The 4$^{th}$ and final iteration when using a subsampling rate of 80 produces the same number of labeled images as the 1$^{st}$ and final iteration when using a subsampling rate of 10, for a fair comparison. Regardless of the number of iterations performed for each subsampling rate, we analyze the time taken to label the full dataset at each iteration. Each uncertainty measure is compared with the randomly selected frames for comparison. A positive time difference shows that the chosen uncertainty measure performed worse than if we were to randomly select frames instead.

The 1$^{st}$ and only iteration for a subsampling rate of 10 can be seen in Table 2, with the same setup outlined in Figure 4. Out of the 3 uncertainty measures applied, selecting the frame with the lowest average confidence value worked the best. However, all 3 of these measures performed worse than randomly selecting frames, although the difference is relatively small. When training the initial model with a subsampling rate of 20, 1 iteration and 2 iterations of active learning were explored. The higher subsampling rate means we are training the initial model with fewer frames, allowing us to increase the number of iterations. When performing only 1 iteration, the setup is similar to that detailed in Figure 4 and the results can be found in Table 3. However, when performing 2 iterations, the setup would be exactly the one shown in Figure 10, with the results expressed in Table 4.

| Uncertainty | Initial | Scan | Add | Remove | Total (hours) | Difference |
|---|---|---|---|---|---|---|
| Random | 28.09 | 12.2 | 6.99 | 1.72 | 49 | 0 |
| **Average** | **28.09** | **12.2** | **7.45** | **1.63** | **49.37** | **0.37** |
| Median | 28.09 | 12.2 | 7.48 | 1.77 | 49.54 | 0.54 |
| Lowest | 28.09 | 12.2 | 7.56 | 1.71 | 49.56 | 0.56 |

*Table 2. Comparing different uncertainty selections with randomly selected frames for the 1st and only active learning iteration for an initial subsampling rate of 10.*

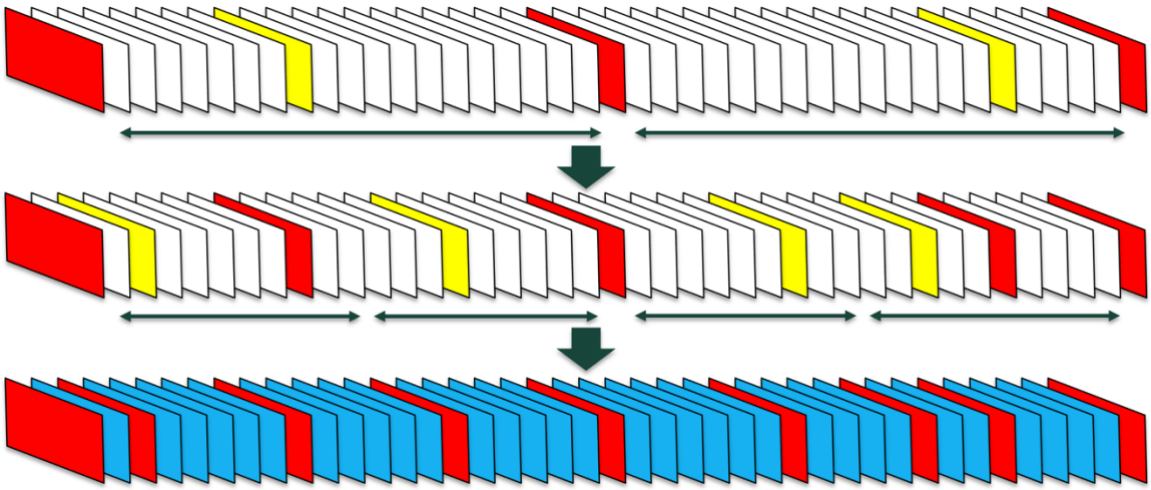

*Figure 10. Performing 2 active learning iterations. The arrows beneath the frames signify the groups that the lowest confidence values are considered from. After both iterations, the selected frames are verified, corrected, and then added to the training set for the next iteration. Lastly, all remaining frames (blue) are tested, verified, and corrected.*

| Uncertainty | Initial | Scan | Add | Remove | Total (hours) | Difference |
|---|---|---|---|---|---|---|
| Random | 14.26 | 12.87 | 10.72 | 2.98 | 40.83 | 0 |
| Average | 14.26 | 12.87 | 11.09 | 2.53 | 40.75 | -0.08 |
| **Median** | **14.26** | **12.87** | **10.96** | **2.64** | **40.73** | **-0.1** |
| Lowest | 14.26 | 12.87 | 11.43 | 2.59 | 41.15 | 0.32 |

*Table 3. Comparing different uncertainty selections with randomly selected frames for the 1st of 2 active learning iterations for an initial subsampling rate of 20.*

It is interesting to note that the 1st iteration for a subsampling rate of 20 produces smaller differences when compared to the random selection than the 2nd iteration. This theme continues when we perform 3 iterations with an initial subsampling rate of 40, as shown in Table 5, Table 6, and Table 7. The 1st iteration for all uncertainty measures improves greatly over the random selection. However, as we perform more iterations, we can see the difference between the random selections becoming minimal.

| Uncertainty | Initial | Scan | Add | Remove | Total (hours) | Difference |
|---|---|---|---|---|---|---|
| Random | 14.26 | 12.87 | 8.11 | 2.19 | 37.43 | 0 |
| Average | 14.26 | 12.87 | 8.86 | 1.83 | 37.82 | 0.39 |
| **Median** | **14.26** | **12.87** | **8.48** | **1.9** | **37.51** | **0.08** |
| Lowest | 14.26 | 12.87 | 8.79 | 1.9 | 37.82 | 0.39 |

*Table 4. Comparing different uncertainty selections with randomly selected frames for the 2nd and final active learning iteration for an initial subsampling rate of 20.*

| Uncertainty | Initial | Scan | Add | Remove | Total (hours) | Difference |
|---|---|---|---|---|---|---|
| Random | 7.08 | 13.21 | 18.3 | 3.27 | 41.86 | 0 |
| Average | 7.08 | 13.21 | 16.3 | 3.56 | 40.15 | -1.71 |
| Median | 7.08 | 13.21 | 15.99 | 3.41 | 39.69 | -2.17 |
| **Lowest** | **7.08** | **13.21** | **15.75** | **3.63** | **39.67** | **-2.19** |

*Table 5. Comparing different uncertainty selections with randomly selected frames for the 1st of 3 active learning iterations for an initial subsampling rate of 40.*

As the subsampling rate increases, selecting frames based on their confidence values significantly decreases the time taken to label the dataset. However, selecting frames based on their confidence values becomes redundant as more iterations are performed. There is little difference between selecting random frames because as we perform more iterations, the frames

become close enough together that there is little difference between them. So selecting specific frames doesn't benefit the model too much as all frames are similar and provide the same information.

| Uncertainty | Initial | Scan | Add | Remove | Total (hours) | Difference |
|---|---|---|---|---|---|---|
| Random | 7.08 | 13.21 | 12.4 | 3.12 | 35.81 | 0 |
| Average | 7.08 | 13.21 | 11.87 | 2.83 | 34.99 | -0.82 |
| **Median** | **7.08** | **13.21** | **11.6** | **2.74** | **34.63** | **-1.18** |
| Lowest | 7.08 | 13.21 | 12.28 | 2.63 | 35.2 | -0.61 |

*Table 6. Comparing different uncertainty selections with randomly selected frames for the 2nd of 3 active learning iterations for an initial subsampling rate of 40.*

| Uncertainty | Initial | Scan | Add | Remove | Total (hours) | Difference |
|---|---|---|---|---|---|---|
| Random | 7.08 | 13.21 | 9.48 | 2.12 | 31.89 | 0 |
| Average | 7.08 | 13.21 | 9.64 | 2.06 | 31.99 | 0.1 |
| **Median** | **7.08** | **13.21** | **9.3** | **2.2** | **31.79** | **-0.1** |
| Lowest | 7.08 | 13.21 | 9.88 | 2.12 | 32.29 | 0.4 |

*Table 7. Comparing different uncertainty selections with randomly selected frames for the 3rd and final active learning iteration for an initial subsampling rate of 40.*

# CHAPTER 4: MIN-DIAL AND MAX-DIAL

Since we are exploiting intra-video training and testing, we developed *Minimal-Distance Iterative Active Learning* (min-DIAL) and *Maximal-Distance Iterative Active Learning* (max-DIAL) approaches. These approaches do not use utilize any confidence values from detections but merely select frames based on their relative location to the anchor frames used to train our model. These methods are much simpler than calculating the average, median, or lowest confidence value for each frame.

## *4.1 Min-DIAL*

Under min-DIAL, after manually labeling an initial subset using a specific subsampling rate, we train our model before testing on the frames closest to the anchor frames. For example, if we trained our model using the $20^{th}$ and $40^{th}$ frames, we would test our model on the frames either side of the $20^{th}$ and the $40^{th}$ frames. This method was chosen due to the frames on either side of the anchor frames being the most similar. Therefore, the outputs will be the most accurate, and fewer corrections are likely to be required. At each iteration, we are updating the model as we propagate inwards. The visualization of the min-DIAL method can be seen in Figure 11.

## *4.2 Max-DIAL*

Under max-DIAL, after manually labeling an initial subset using a specific subsampling rate, we train our model before testing on the next subsampling rate down. For example, if we trained our model using every 80th frame (including the 0th frame), we would then test on every 40th frame that wasn't included in the training set. This method was chosen based on the fact that the unlabeled frames that sit centrally between two anchor frames are likely to be the most uncertain from that particular group. This is based on the fact that they are the furthest distance

away from any frames used to train the model. After manually verifying and correcting any incorrect detections, we are left with the ground truth labels for every 40th frame. The selection of frames using the max-DIAL approach can be visualized in Figure 12.

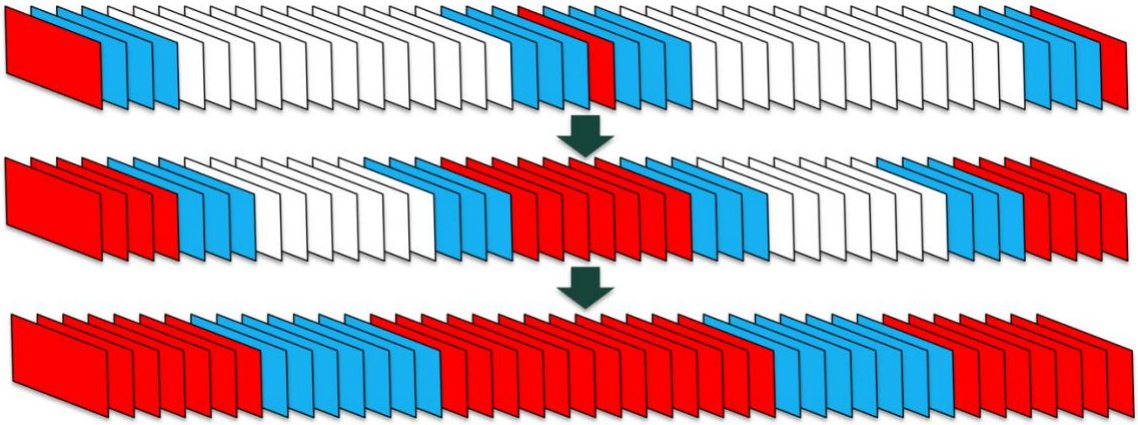

*Figure 11. Overview of the min-DIAL method. The trained model is tested on the frames closest to either side of the anchor frames. After each iteration, the frames tested are verified, corrected, and added to the training set for the next iteration.*

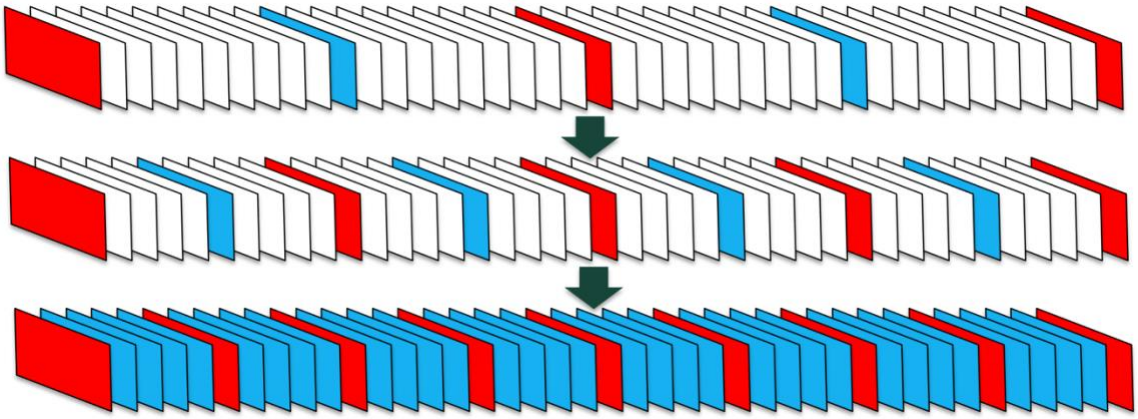

*Figure 12. Overview of the max-DIAL method. The trained model is tested on the frames that sit centrally between the anchor frames. After each iteration, the frames that are tested are verified, corrected, and added to the training set for the next iteration.*

To see which method works best, we compare the time taken to label the full dataset using different subsampling frequencies. Table 8 shows the different times using min-DIAL and max-DIAL and Figure 13 provides a graphical view of the time taken at each subsampling frequency. From Table 8 and Figure 13, we can see that max-DIAL performs the best out of the

two proposed methods. In fact, as the subsampling frequency increases, the distance between the times for each method increases. A potential reason for this could be as we increase the initial subsampling frequency, min-DIAL requires more neighboring frames to be tested on at each iteration to compensate for the larger number of unlabeled frames between anchor frames.

| Frequency | Initial | Scan | Add | Remove | Total (hours) |
|---|---|---|---|---|---|
| 5 | 55.72 | 10.84 | 5.58 | 1.19 | 73.33 |
| 10 + 1xMin-DIAL | 28.09 | 12.19 | 7.03 | 1.84 | 49.15 |
| 10 + 1xMax-DIAL | 28.09 | 12.2 | 6.52 | 1.6 | 48.41 |
| 20 + 2xMin-DIAL | 14.26 | 12.87 | 9.03 | 1.85 | 38.01 |
| 20 + 2xMax-DIAL | 14.26 | 12.87 | 7.37 | 1.86 | 36.36 |
| 40 + 3xMin-DIAL | 7.08 | 13.21 | 11.5 | 2.3 | 34.09 |
| 40 + 3xMax-DIAL | 7.08 | 13.21 | 8.3 | 1.95 | 30.54 |
| 80 + 4xMin-DIAL | 3.51 | 13.38 | 15.73 | 2.8 | 35.42 |
| 80 + 4xMax-DIAL | 3.51 | 13.38 | 8.8 | 2.01 | 27.7 |

*Table 8. Time taken to completely label the dataset using different initial subsampling frequencies and different numbers of min-DIAL and max-DIAL iterations.*

Moving forward, we will only consider Max-DIAL due to its superiority. Table 9 outlines the full range of subsampling frequencies for our max-DIAL method. Note that we add training times because as we increase the subsampling frequency, which subsequently increases the number of possible active learning iterations, there becomes a point where minimal time is saved by increasing the initial subsampling frequency. By including the training times, we can show that the minimal time decrease becomes redundant with the extra time taken to train the model for another iteration. The effect of adding the training time can be better expressed in Figure 14.

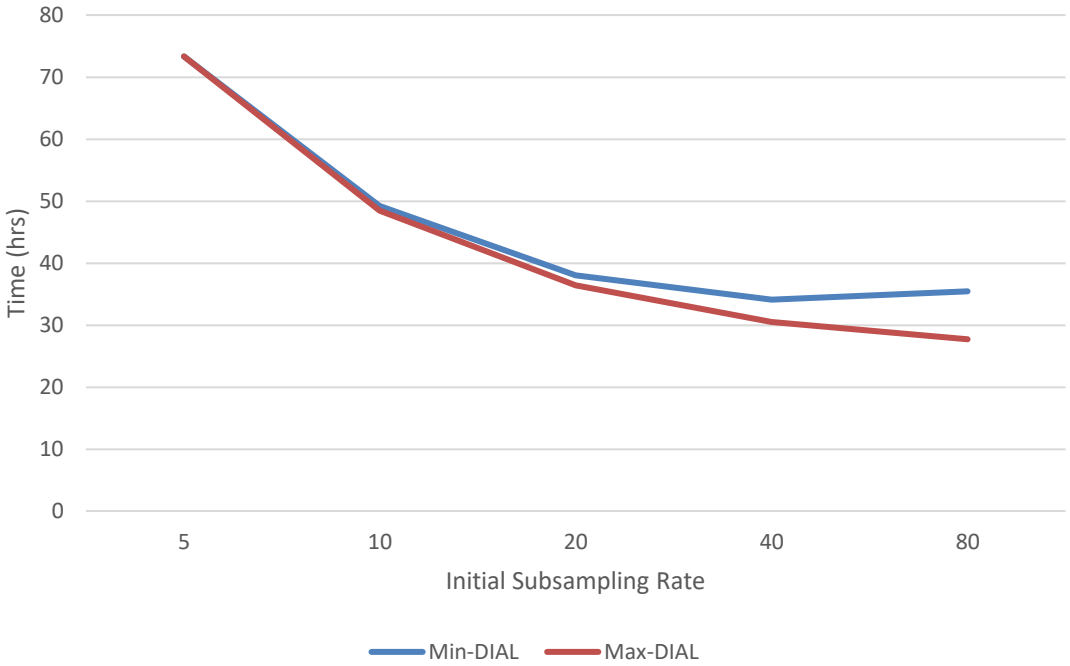


*Figure 13. Visual representation of the total time taken to label the full dataset using min-DIAL and max-DIAL using different subsampling frequencies.*

| Frequency | Initial | Scan | Add | Remove | Total (hours) | Train | Complete (hours) |
|---|---|---|---|---|---|---|---|
| 5 | 55.72 | 10.84 | 5.58 | 1.19 | 73.33 | 3 | 76.33 |
| 10 + 1xAL | 28.09 | 12.2 | 6.52 | 1.6 | 48.41 | 6 | 54.41 |
| 20 + 2xAL | 14.26 | 12.87 | 7.37 | 1.86 | 36.36 | 9 | 45.36 |
| 40 + 3xAL | 7.08 | 13.21 | 8.3 | 1.95 | 30.54 | 12 | 42.54 |
| 80 + 4xAL | 3.51 | 13.38 | 8.8 | 2.01 | 27.7 | 15 | 42.7 |
| 160 + 5xAL | 1.61 | 13.47 | 9.09 | 2.06 | 26.23 | 18 | 44.23 |
| 320 + 6xAL | 0.72 | 13.51 | 9.22 | 2.07 | 25.52 | 21 | 46.52 |
| 640 + 7xAL | 0.31 | 13.53 | 9.31 | 2.07 | 25.22 | 24 | 49.22 |
| 1280 + 8xAL | 0.03 | 13.54 | 9.4 | 2.08 | 25.05 | 27 | 52.05 |

*Table 9. Different initial subsampling frequencies with our max-DIAL method. The total time refers to the amount of human time required to label the dataset. The complete time is the total human time but also accounts for the model's training time.*

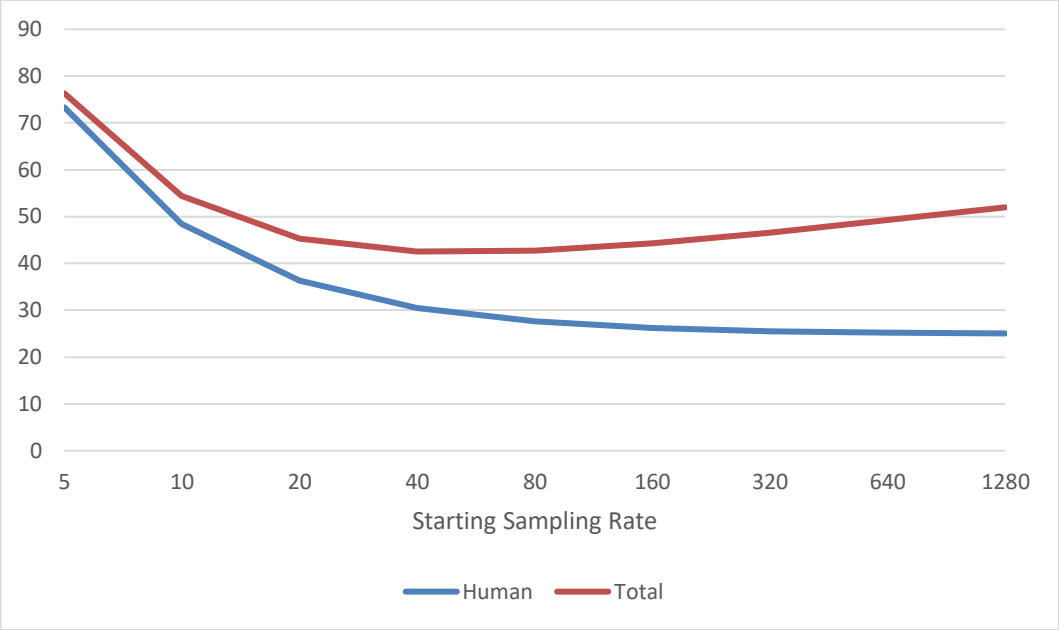


*Figure 14. Max-DIAL method using different initial subsampling frequencies. Total time considers training time also. Although increasing the initial subsampling rate decreases the time taken, after a subsampling rate of 80, the time begins to level out.*

### *4.3 Comparing Max-DIAL with Traditional Active Learning*

The results for an initial subsampling rate of 10, with one iteration of active learning, can be seen in Table 10. The max-DIAL results are compared to the randomly selected frames and the most efficient uncertainty measure from the traditional active learning methods. Even when performing 1 iteration, our max-DIAL method decreases the time taken to label the dataset. When applying our max-DIAL method to a subsampled frequency of 20, we continue to see promising improvements over traditional active learning methods. Table 11 and Table 12 outline the times for performing 1 and 2 iterations, respectively, using our max-DIAL method. The 3 iterations for an initial subsampling frequency of 40 can be seen in Tables 13, 14, and 15. At every iteration, across all subsampling frequencies, max-DIAL outperforms the random selection of frames. It also outperforms or performs equivalently to all other uncertainty measures. Following previous patterns, the initial iterations produce a larger gap in performance over the

randomly selected sample. As we increase the number of iterations, the difference becomes smaller as we incorporate more and more frames.

| Uncertainty | Initial | Scan | Add | Remove | Total (hours) | Difference |
|---|---|---|---|---|---|---|
| Random | 28.09 | 12.2 | 6.99 | 1.72 | 49 | 0 |
| Average | 28.09 | 12.2 | 7.45 | 1.63 | 49.37 | 0.37 |
| **Max-DIAL** | **28.09** | **12.2** | **6.52** | **1.7** | **48.51** | **-0.49** |

*Table 10. Comparing random selection and the most efficient traditional active learning technique previously computed (average) with our max-DIAL method for the 1st and only iteration for an initial subsampling rate of 10.*

| Uncertainty | Initial | Scan | Add | Remove | Total (hours) | Difference |
|---|---|---|---|---|---|---|
| Random | 14.26 | 12.87 | 10.72 | 2.98 | 40.83 | 0 |
| Median | 14.26 | 12.87 | 10.96 | 2.64 | 40.73 | -0.1 |
| **Max-DIAL** | **14.26** | **12.87** | **10.13** | **2.64** | **39.9** | **-0.93** |

*Table 11. Comparing random selection and the most efficient traditional active learning technique previously computed (median) with our max-DIAL method for the 1st of 2 iterations for an initial subsampling rate of 20.*

| Uncertainty | Initial | Scan | Add | Remove | Total (hours) | Difference |
|---|---|---|---|---|---|---|
| Random | 14.26 | 12.87 | 8.11 | 2.19 | 37.43 | 0 |
| Median | 14.26 | 12.87 | 8.48 | 1.9 | 37.51 | 0.08 |
| **Max-DIAL** | **14.26** | **12.87** | **7.9** | **1.87** | **36.9** | **-0.53** |

*Table 12. Comparing random selection and the most efficient traditional active learning technique previously computed (median) with our max-DIAL method for the 2nd and final iteration for an initial subsampling rate of 20.*

| Uncertainty | Initial | Scan | Add | Remove | Total (hours) | Difference |
|---|---|---|---|---|---|---|
| Random | 7.08 | 13.21 | 18.3 | 3.27 | 41.86 | 0 |
| **Single** | **7.08** | **13.21** | **15.75** | **3.63** | **39.67** | **-2.19** |
| Max-DIAL | 7.08 | 13.21 | 16 | 3.59 | 39.88 | -1.98 |

*Table 13. Comparing random selection and the most efficient traditional active learning technique previously computed (single) with our max-DIAL method for the 1st of 3 iterations for an initial subsampling rate of 40.*

| Uncertainty | Initial | Scan | Add | Remove | Total (hours) | Difference |
|---|---|---|---|---|---|---|
| Random | 7.08 | 13.21 | 12.4 | 3.12 | 35.81 | 0 |
| Median | 7.08 | 13.21 | 11.6 | 2.74 | 34.63 | -1.18 |
| **Max-DIAL** | **7.08** | **13.21** | **11.16** | **2.68** | **34.13** | **-1.68** |

*Table 14. Comparing random selection and the most efficient traditional active learning technique previously computed (median) with our max-DIAL method for the 2nd of 3 iterations for an initial subsampling rate of 40.*

| Uncertainty | Initial | Scan | Add | Remove | Total (hours) | Difference |
|---|---|---|---|---|---|---|
| Random | 7.08 | 13.21 | 9.48 | 2.12 | 31.89 | 0 |
| Median | 7.08 | 13.21 | 9.3 | 2.2 | 31.79 | -0.1 |
| **Max-DIAL** | **7.08** | **13.21** | **8.93** | **1.96** | **31.18** | **-0.71** |

*Table 15. Comparing random selection and the most efficient traditional active learning technique previously computed (median) with our max-DIAL method for the 3rd and final iteration for an initial subsampling rate of 40.*

# CHAPTER 5: JENSEN-SHANNON DIVERGENCE

After training a model with uniformly sampled anchor frames, more advanced methods can be explored to select uncertain frames to be used in the next iteration. The Jensen-Shannon Divergence (JSD) calculates the distance between two distributions, P and Q, shown in Equation 4. The JSD distance metric is based on the Kullback-Leibler divergence between 2 distributions, KL(P || Q), but is symmetric, meaning that the order of the distributions is irrelevant. The output is between 0 and 1, with 0 showing no difference between the two distributions.

$$JSD(P \,||\, Q) = \frac{1}{2}KL(P \,||\, M) + \frac{1}{2}KL(Q \,||\, M), \text{ where } M = \frac{1}{2}(P + Q) \quad (4)$$

If we create a distribution for each frame, we can select the frames of interest based on the difference between the anchor frames. The frames with a larger difference from the anchor frames have a higher probability of providing the model with more information. As these frames will contain different objects, or the same objects but located in different parts of the frame, adding these frames to the training set will likely improve the model the most.

## *5.1 Creating Distributions*

There are many ways to create a distribution for an image. Creating histograms based on the number of different colored pixel values is one option. However, many of the frames in our video sequences may contain completely different backgrounds but include the same objects. As we are more interested in the objects in the frame, this technique is not beneficial for our application. Our method for creating histograms is to divide each frame into smaller sections and count the number of detections in each subsection, shown in Figure 15. Based on the fact that the detections are highly accurate using the intra-video training and testing method, the majority of the detections are correct and can be considered as actual objects, with the distribution shown in Figure 16. If we divide each frame into 2 or 3 sections, there is not much difference in the

distributions because the locations of the detections have to change significantly. Conversely, if we divide the frame into a large number of sections, almost every frame that is compared, even if they are neighboring frames, receive a high difference value. This is because an object only has to move a small distance to enter another section, meaning the distribution will change almost every frame. For this reason, we used 8 different sections as they produce reasonably wide sections. We use ground truth labels to calculate the distributions for the anchor frames.

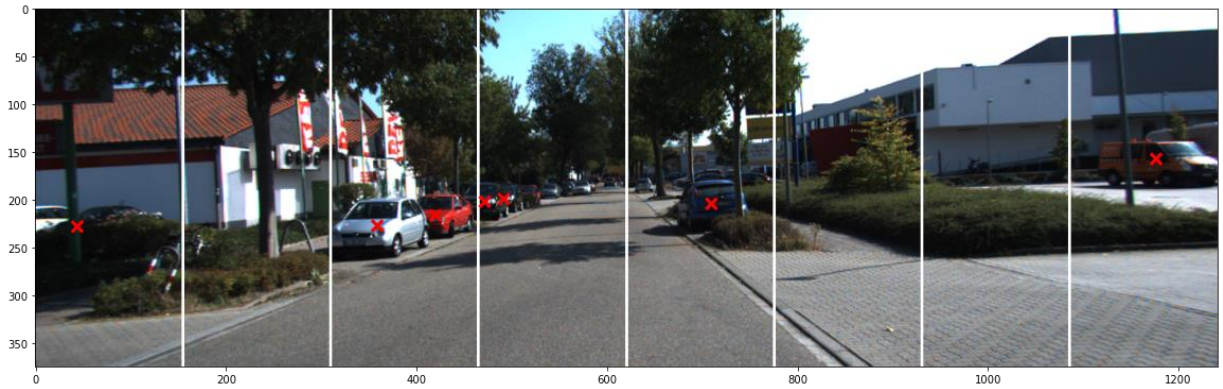


*Figure 15. A frame divided into 8 sections containing multiple detections. The center point of the bounding box for each detection is marked as a red cross. Each center location is placed in one of the 8 bins based on which section it falls in.*

### *5.2 Calculating Difference Value*

When calculating a difference value for a frame, it is important to incorporate the anchor frames from either side. For example, if we wanted to select a frame from between the 10$^{th}$ and 20$^{th}$ frames of a video sequence, we need to calculate each frame's difference from both the 10$^{th}$ and the 20$^{th}$ frames. It is clear that the 11$^{th}$ frame will share the most similarities with the lower anchor frame (the 10$^{th}$ frame) and be most different from the upper anchor frame (the 20$^{th}$ frame). So, calculating the difference between a frame and only one of the anchor frames doesn't provide much information. Therefore, we compute the average of the differences between each frame and the lower and upper anchor frames. By averaging the two differences, we are more

likely to select frames from around the middle, where the frames are different from both anchor frames. If an anchor frame has no ground truth labels or a frame from between the two anchor frames we are comparing with has no detections, the difference score from the two frames is given a value of zero. If all frames between two anchor frames have the same difference value, we select the middle frame. This is based on the success shown using our max-DIAL method.

*5.3 JSD Results and Comparison*

The results for selecting the frames to use for the next iteration based on our JSD uncertainty measure compared with all previous active learning methods for a subsampled rate of 10 can be seen in Table 16. Although our JSD approach takes slightly less time than the traditional methods, it doesn't perform as well as our max-DIAL approach.

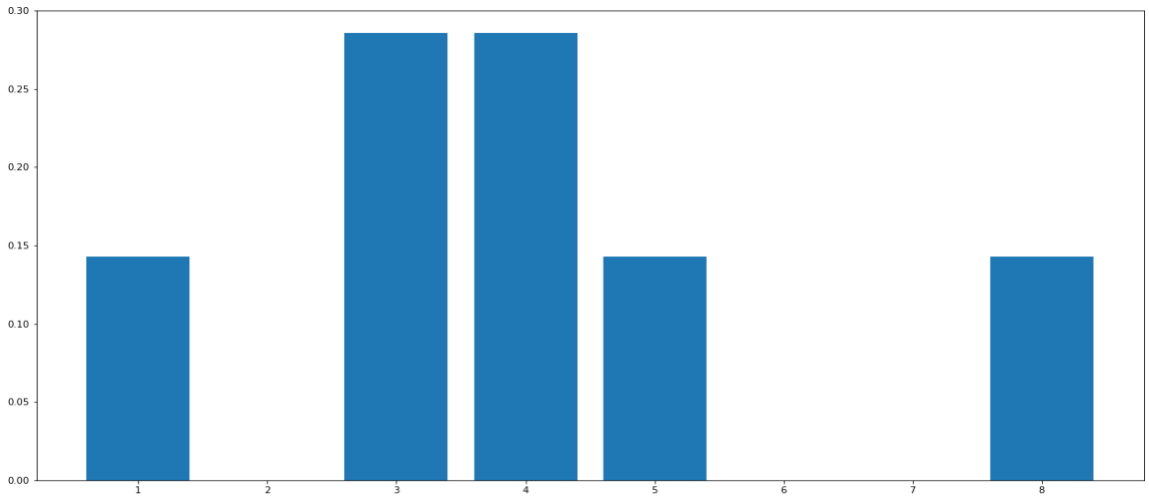


*Figure 16. The distribution of the center points for each detection from Figure 15.*

Extending our JSD method to a subsampling rate of 20, we can see the results for 1 and 2 iterations in Tables 17 and 18. Our JSD method now performs slightly worse than the traditional active learning method. However, our max-DIAL approach still performs the best out of all approaches. Lastly, we use our JSD uncertainty measure to select frames from the 3 iterations

performed using a subsampling rate of 40. The results are shown in Figures 19, 20, and 21, respectively. The JSD method appears to perform similarly to traditional active learning methods. The difference is often minimal, showing that the JSD method isn't a very effective method for selecting uncertain frames. Again, however, our max-DIAL method performs substantially better than all other active learning methods.

| Uncertainty | Initial | Scan | Add | Remove | Total (hours) | Difference |
|---|---|---|---|---|---|---|
| Random | 28.09 | 12.2 | 6.99 | 1.72 | 49 | 0 |
| Average | 28.09 | 12.2 | 7.45 | 1.63 | 49.37 | 0.37 |
| Median | 28.09 | 12.2 | 7.48 | 1.77 | 49.54 | 0.54 |
| Single | 28.09 | 12.2 | 7.56 | 1.71 | 49.56 | 0.56 |
| **Max-DIAL** | **28.09** | **12.2** | **6.52** | **1.7** | **48.51** | **-0.49** |
| JSD | 28.09 | 12.2 | 6.97 | 1.79 | 49.05 | 0.05 |

*Table 16. Comparison of JSD with all other active learning methods for the 1st and only iteration for an initial subsampling rate of 10.*

| Uncertainty | Initial | Scan | Add | Remove | Total (hours) | Difference |
|---|---|---|---|---|---|---|
| Random | 14.26 | 12.87 | 10.72 | 2.98 | 40.83 | 0 |
| Average | 14.26 | 12.87 | 11.09 | 2.53 | 40.75 | -0.08 |
| Median | 14.26 | 12.87 | 10.96 | 2.64 | 40.73 | -0.1 |
| Single | 14.26 | 12.87 | 11.43 | 2.59 | 41.15 | 0.32 |
| **Max-DIAL** | **14.26** | **12.87** | **10.13** | **2.64** | **39.9** | **-0.93** |
| JSD | 14.26 | 12.87 | 11.77 | 2.55 | 41.45 | 0.62 |

*Table 17. Comparison of JSD with all other active learning methods for the 1$^{st}$ of 2 iterations for an initial subsampling rate of 20.*

| Uncertainty | Initial | Scan | Add | Remove | Total (hours) | Difference |
|---|---|---|---|---|---|---|
| Random | 14.26 | 12.87 | 8.11 | 2.19 | 37.43 | 0 |
| Average | 14.26 | 12.87 | 8.86 | 1.83 | 37.82 | 0.39 |
| Median | 14.26 | 12.87 | 8.48 | 1.9 | 37.51 | 0.08 |
| Single | 14.26 | 12.87 | 8.79 | 1.9 | 37.82 | 0.39 |
| **Max-DIAL** | **14.26** | **12.87** | **7.9** | **1.87** | **36.9** | **-0.53** |
| JSD | 14.26 | 12.87 | 8.85 | 1.94 | 37.92 | 0.49 |

*Table 18. Comparison of JSD with all other active learning methods for the 2nd and final iteration for an initial subsampling rate of 20.*

| Uncertainty | Initial | Scan | Add | Remove | Total (hours) | Difference |
|---|---|---|---|---|---|---|
| Random | 7.08 | 13.21 | 18.3 | 3.27 | 41.86 | 0 |
| Average | 7.08 | 13.21 | 16.3 | 3.56 | 40.15 | -1.71 |
| Median | 7.08 | 13.21 | 15.99 | 3.41 | 39.69 | -2.17 |
| **Single** | **7.08** | **13.21** | **15.75** | **3.63** | **39.67** | **-2.19** |
| Max-DIAL | 7.08 | 13.21 | 16 | 3.59 | 39.88 | -1.98 |
| JSD | 7.08 | 13.21 | 16.25 | 3.67 | 40.21 | -1.65 |

*Table 19. Comparison of JSD with all other active learning methods for the 1st of 3 iterations for an initial subsampling rate of 40.*

| Uncertainty | Initial | Scan | Add | Remove | Total (hours) | Difference |
|---|---|---|---|---|---|---|
| Random | 7.08 | 13.21 | 12.4 | 3.12 | 35.81 | 0 |
| Average | 7.08 | 13.21 | 11.87 | 2.83 | 34.99 | -0.82 |
| Median | 7.08 | 13.21 | 11.6 | 2.74 | 34.63 | -1.18 |
| Single | 7.08 | 13.21 | 12.28 | 2.63 | 35.2 | -0.61 |
| **Max-DIAL** | **7.08** | **13.21** | **11.16** | **2.68** | **34.13** | **-1.68** |
| JSD | 7.08 | 13.21 | 12.77 | 2.8 | 35.86 | 0.05 |

*Table 20. Comparison of JSD with all other active learning methods for the 2nd of 3 iterations for an initial subsampling rate of 40.*

| Uncertainty | Initial | Scan | Add | Remove | Total (hours) | Difference |
|---|---|---|---|---|---|---|
| Random | 7.08 | 13.21 | 9.48 | 2.12 | 31.89 | 0 |
| Average | 7.08 | 13.21 | 9.64 | 2.06 | 31.99 | 0.1 |
| Median | 7.08 | 13.21 | 9.3 | 2.2 | 31.79 | -0.1 |
| Single | 7.08 | 13.21 | 9.88 | 2.12 | 32.29 | 0.4 |
| **Max-DIAL** | **7.08** | **13.21** | **8.93** | **1.96** | **31.18** | **-0.71** |
| JSD | 7.08 | 13.21 | 9.32 | 2.12 | 31.73 | -0.16 |

*Table 21. Comparison of JSD with all other active learning methods for the 3rd and final iteration for an initial subsampling rate of 40.*

# CONCLUSION

We have introduced a semi-automated video labeling framework that attempts to minimize human interaction time while applying active learning strategies to maximize the accuracy of our automated labeling process. We have shown that applying our proposed SALV framework, which exploits training, testing, and active learning on frames from the same video sequences, produces highly accurate results. This is due to the similarity of the environment between images captured within a small timeframe. Combining intra-video sequence training and testing with our max-DIAL approach for active learning, we further improved the accuracy of the detections and reduced the time taken to label the full dataset. Our max-DIAL approach outperformed all the traditional active learning methods explored, as well as our proposed JSD approach, allowing us to reduce the labeling time by more than 90%, compared to manual labeling. It is important to note that we did not include the training times of the model in our calculations. This is based on the fact that training does not require any assistance from a human. While the training is running, other tasks for semi-automated labeling can be completed. We are also more interested in reducing the workload on the human; therefore, we are only interested in how much time the human needs to spend on the overall labeling process.

Future work could look at an ablation study that varies the confidence threshold of the detections. In this thesis we allowed all detections to be present. However, only allowing high-confidence detections would improve the precision, meaning fewer detections need to be removed. Conversely, this may reduce the recall, meaning that more bounding boxes have to be added, which is more time-consuming than removing them. An optimum confidence threshold could potentially reduce the time even further.

## BIBLIOGRAPHY


[1] J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li, and Li Fei-Fei, ”ImageNet: A large-scale hierarchical image database,” 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.

[2] K. Crowston, ”Amazon Mechanical Turk: A Research Tool for Organizations and Information Systems Scholars,” IFIP Advances in Information and Communication Technology, vol 389, 2012. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642 35142-6 14.

[3] L. Fei-Fei. (2010). *ImageNet. Crowdsourcing, benchmarking other cool things* [Online]. Available: https://www.image- net.org/static files/papers/ImageNet2010.pdf.

[4] T. A. Biresaw, T. Nawaz, J. Ferryman, and A. I. Dell, ”ViTBAT: Video tracking and behavior annotation tool,” 2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Colorado Springs, CO, USA, 2016, pp. 295-301, doi: 10.1109/AVSS.2016.7738055.

[5] D. P. Papadopoulos, J. R. R. Uijlings, F. Keller and V. Ferrari, ”Training Object Class Detectors with Click Supervision,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 180-189, doi: 10.1109/CVPR.2017.27.

[6] D. Li, J. -B. Huang, Y. Li, S. Wang and M. -H. Yang, ”Weakly Supervised Object Localization with Progressive Domain Adaptation,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 3512-3520, doi: 10.1109/CVPR.2016.382.

[7] D. P. Papadopoulos, J. R. R. Uijlings, F. Keller and V. Ferrari, ”We Don’t Need No Bounding-Boxes: Training Object Class Detectors Using Only Human Verification,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 854-863, doi: 10.1109/CVPR.2016.99.

[8] D. P. Papadopoulos, J. R. R. Uijlings, F. Keller and V. Ferrari, ”Extreme Clicking for Efficient Object Annotation,” 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 4940-4949, doi: 10.1109/ICCV.2017.528.

[9] D.Schorkhuber, F.Grohand, and M.Gelautz, ”Bounding Box Propagation for Semi-automatic Video Annotation of Nighttime Driving Scenes,” 2021 12th International Symposium on Image and Signal Processing and Analysis (ISPA), Zagreb, Croatia, 2021, pp. 131-137, doi: 10.1109/ISPA52656.2021.9552141.

[10] B. -L. Wang, C. -T. King and H. -K. Chu, ”A Semi-Automatic Video Labeling Tool for Autonomous Driving Based on Multi-Object Detector and Tracker,” 2018 Sixth International Symposium on Computing and Networking (CANDAR), Takayama, Japan, 2018, pp. 201-206, doi: 10.1109/CANDAR.2018.00035.

[11] B. Adhikari, J. Peltomaki, J. Puura, and H. Huttunen, ”Faster Bounding Box Annotation for Object Detection in Indoor Scenes,” 2018 7th European Workshop on Visual Information Processing (EUVIP), Tampere, Finland, 2018, pp. 1-6, doi: 10.1109/EUVIP.2018.8611732.

[12] C. -A. Brust, C. Ka¨ding, and J. Denzler, ”Active learning for deep object detection,” arXiv preprint arXiv:1809.09875, 2018.

[13] T. Yuan et al., ”Multiple Instance Active Learning for Object Detection,” 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2021, pp. 5326-5335, doi: 10.1109/CVPR46437.2021.00529.

[14] J. Choi, I. Elezi, H. -J. Lee, C. Farabet and J. M. Alvarez, ” Active Learning for Deep Object Detection via Probabilistic Modeling,” 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021, pp. 10244-10253, doi: 10.1109/ICCV48922.2021.01010.

[15] J. Zolfaghari Bengar et al., ”Temporal Coherence for Active Learning in Videos,” 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea (South), 2019, pp. 914-923, doi: 10.1109/ICCVW.2019.00120.

[16] G. Jocher et al. YOLOv5, v7.0, doi: 10.5281/zenodo.3908559 [online]. Available: https://github.com/ultralytics/yolov5.

[17] A. Geiger, P. Lenz, and R. Urtasun, ”Are we ready for autonomous driving? The KITTI vision benchmark suite,” 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 2012, pp. 3354-3361, doi: 10.1109/CVPR.2012.6248074.

[18] H. Su, J. Deng, and L. Fei-Fei. *Crowdsourcing Annotations for Visual Object Detection*. [Online]. Available: http://vision.stanford.edu/pdf/bbox submission.pdf.