Towards Efficient Frame Sampling Strategies for Video Action Recognition

Tanushree Banerjee '24, Ameya Vaidya '24, Brian Lou '24
Advisors: Jihoon Chung, Prof Olga Russakovsky
COS429: Computer Vision, Spring 2023
Final Project Written Report

Abstract

Most video action recognition models use raw videos as input [36]. However, not only is this approach inefficient since video data has high temporal redundancy, but also infeasible in practice due to computational and storage constraints [36]. Thus, in practice, video action recognition models sample a subset of all frames in a video in order to reduce computational expense and inference time. Frame sampling strategies that are able to select the most salient and/or discriminative frames from a video would be most efficient at reducing the required computational expense by reducing the number of frames that need to be further processed by an action recognition model. We hypothesize that the norm of the optical flow vector of a frame and the number of objects in a frame provide a signal for which frames could be the most salient for video action recognition. Based on these hypotheses, we propose and evaluate two dataset and model-agnostic frame sampling strategies: one based on the norm of the optical flow of frames, and the other based on the number of objects in frames. We find that although our hypotheses hold, i.e. there is some additional signal provided by the optical flow norm and number of objects in the frame about whether the frame is salient for action recognition, our proposed approaches do not perform considerably better than some of the less computationally expensive baselines introduced later in the paper. Code is available at this GitHub repository.

1. Motivation and Goal

Video footage constitutes a significant proportion of all digital data available [1], and has a tremendous capacity to encapsulate useful information about the world [1]. Once extracted from raw video data, this information could be used to build models that could revolutionize manufacturing [1], help build smart cities [1] and autonomous vehicles [1], amongst other high-stakes applications that would make several tasks much more cost-effective and efficient [1]. Thus, the abundance, richness, and applicability of video data makes video understanding a key challenge in computer vision [1], and a hot area for research [1].

As opposed to tasks based on static image processing, video understanding tasks require the consideration of temporal information along with much more visual information as input. Thus, video understanding models are much more complex and computationally expensive than traditional static image-based models [34], making these tasks much more challenging to solve while ensuring such models are usable given the computational constraints in practice.

Existing research on visual understanding tasks has mostly focused on obtaining compact yet effective video representations for efficient and robust recognition [34]. However, figuring out an efficient inference strategy for fast processing is also essential for video understanding models to be usable in practical applications [34]. While the high time redundancy in video data makes it unnecessary to to feed entire videos into a model for subsequent processing, computational budget constraints in practice often make this infeasible [34]. Thus, finding an efficient strategy for sampling a small subset of the frames of a video for further processing is a more fundamental challenge in video understanding - one that remains to be unsolved [34]. Therefore, it is crucial to investigate efficient sampling strategies for video understanding tasks in order to develop models that can be used in practice.

The canonical approach for sampling frames typically involves employing a fixed hand-crafted sampling strategy for training and testing in videos [20, 23, 25]. The model is trained on frames/clips that are randomly sampled either evenly or successively with a fixed stride from the original video [34]. During the test phase, in order to cover the full duration of the video, clips are densely sampled from the video, and the final output is determined by averaging these dense prediction scores [34].

Yet, there are several problems associated with such fixed, hand-crafted sampling strategies. Firstly, since the exact set of frames where the action occurs in a video is not fixed across different videos, fixing which frames are sam-

pled across videos may cause actions in frames that happen to not be sampled to be missed by the model. Moreover, not all frames are equally salient for video understanding tasks, and an efficient sampling strategy should attempt to select more discriminative frames rather than irrelevant background frames that do not meaningfully affect predictions or provide additional new information compared to other selected frames.

Prior work on devising efficient sampling strategies have involved adding an adaptive sampling module [7, 28, 30]. Typically, these modules are trained to select more discriminative frames for subsequent processing [34]. However, such methods heavily depend on the training data, and cannot easily transfer to unseen actions [34]. Thus, further research is required in order to develop sampling strategies that are both task and dataset-agnostic.

In this paper, we aim to present and evaluate simple, sparse, and explainable sampling strategies for video understanding that are (1) more generalizable and independent of the training data, and (2) able to deal with varied video content adaptively at inference time.

In particular, we focus our analysis on the task of video action recognition, which involves recognizing human actions in a video [36]. We focus our analysis on this task since it is one of the most representative tasks for video understanding [36], requiring the model to be able to recognize, localize, and predict human behaviors [36].

Two key ideas motivate the proposed sampling strategies in this paper.

1. Frames with greater apparent motion are more likely to convey new information about human action in a video, and are thus more likely to be salient and/or discriminative for human action prediction. Optical flow quantifies the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene [4]. Thus, the optical flow of a frame may provide a universal and transferable signal for identifying salient and/or discriminative frames.

2. Frames with more objects are more likely to contain human action than background scene frames that do not contain objects. Thus, the number of objects in a frame may also provide another dataset-agnostic and transferable signal for identifying salient and/or discriminative frames. Frames without objects are more likely to be irrelevant or redundant for action recognition.

Thus, more specifically, this paper aims to explore an optical flow-based and object detection-based frame sampling strategy for more resource-efficient video action recognition, along with more simple baseline sampling strategies to compare our proposed strategies against. This paper goes on to thoroughly investigate and analyse the proposed sampling strategies both quantitatively and qualitatively in order to draw insights about which types of sampling strategies might allow us to identify and select the most salient and/or discriminative frames for efficient action recognition and why.

We find that although our hypotheses hold, i.e. there is some additional signal provided by the optical flow norm and number of objects in the frame about whether the frame is salient for action recognition, our proposed approaches do not perform significantly better than some of the less computationally expensive baselines introduced later in the paper.

2. Problem Background and Related Work

2.1. Compressed video action recognition

Inspired by the progress in the video compression domain, several works have adopted compressed video representations as input to train computationally efficient video models.

Video compression methods often store a frame by reusing contents from the original RGB video frame and only store the difference, as captured by the motion vector and residual [36]. These methods rely on the assumption that adjacent frames are most often very similar [36].

2.1.1 Knowledge distillation-based approach

Zhang et al. [32] take into account the coarse structure of the motion vector and the inaccurate movements they may contain by adopting 'knowledge distillation' to help the motion-vector-based temporal stream mimic the optical-flow-based temporal stream [32]. Their approach is 27 times faster than standard two-stream networks while maintaining comparable accuracy. However, their approach requires extracting and processing each frame, making it computationally expensive.

2.1.2 CNN-based approach

Wu et al. [27] use a heavyweight CNN for the original RGB video frame and lightweight CNN's for the motion vector frames obtained. However, this approach requires that the motion vectors and residuals for each frame be referred back to the original RGB frame by accumulation.

DMC-Net [19] follows up on Wu et al. [27] using adversarial loss. They adopt a lightweight generator network to help the motion vector capturing fine motion details, instead of knowledge distillation as in Zhang et al. [32]

2.2. Frame/Clip sampling

Many methods take inspiration from normal human classification which involve skimming over a video and only

using just a few glimpses [31]. These methods involve sampling the most informative video frames to both improve performance and improve model efficiency during inference.

2.2.1 Naive frame sampling strategies

Some 3D CNN based methods [5, 10, 22] use a naive frame sampling strategy which first selects a random frame in the video, then concatenates a subset of the next 64 consecutive frames using uniform sub-sampling. **TSN [26]**, on the other hand, samples frames uniformly.

Both these sampling strategies are often used by action recognition models. However, they treat every frame equally and ignore the redundancy between frames [36], making them inefficient.

2.2.2 Initial salient frame sampling strategies

KVM [35] was one of the earliest attempts at actively selecting salient frames from a video. They do so by using a framework that identifies key frames and performs action recognition simultaneously. On the other hand, **AdaScan [13]** computes an importance score for each frame in an online manner, which is termed "adaptive temporal pooling". However, both these methods are not efficient at inference time [36].

2.2.3 Reinforcement learning-based frame sampling strategies

Recently, several works proposed reinforcement learning (RL) to train agents using policy gradient methods to select frames [36]. **FastForward [18]** uses RL for frame-skipping, planning and early-stop decision making to reduce the computation burden for untrimmed video action recognition. **Adaframe [30]** proposes an LSTM augmented with a global memory to search which frames to use over time, which was trained by policy gradient methods to search more informative video clips. **Multi-agent [28]** uses N agents where each agent selects an informative frame/clip from a video.n **DSN [33]** proposes a dynamic version of TSN which utilizes RL-based sampling. **LiteEval [29]** avoids complicated RL policy gradients by presenting a coarse-to-fine and differentiable framework containing a coarse LSTM and a fine LSTM organized hierarchically, in addition to a gating module for selecting coarse or fine features. **AR-Net [24]** proposes a unified framework for selecting optimal frame resolutions and skipping. This framework is learnt in a fully differentiable manner.

2.2.4 Motion-based frame sampling strategies

MGSampler [34] proposed a motion-uniform sampling strategy that samples frames based on motion distribution. This strategy ensures even coverage of all video segments while maintains high motion salience in frames. They demonstrate that their sampling method achieves higher effectiveness compared to fixed sampling strategies using 5 different benchmarks.

2.2.5 Sparse Sampling

ClipBERT [16] utilized the idea of sparse sampling using only a randomly sampled one-second segment of a video. They found that this sparse sampling technique outperforms similar video and language models which use full-length videos, and also generalizes well between different domains.

2.2.6 Audio-based frame sampling strategies

Audio has also been used as an efficient way to select salient frames for action recognition in several prior works.

SCSampler Korbar et al., 2019 [15] uses a lightweight CNN as the selector which attempts to sample the most salient video clips based on compressed video representations at test time using salience scores. To train the selector, they use audio as an extra input. Their method achieves state-of-the-art performance on both Kinetics400 and Sports1M dataset. They also empirically show that such saliency-based sampling is not only efficient, but also achieves higher accuracy than using all video frames.

Listen to Look [11] uses audio to remove short-term and long-term visually redundant frames for fast video action recognition at inference time.

Although all the aforementioned approaches improve action recognition model performance, the design of their sampling module is often complex and computationally expensive. Moreover, the training process of their sampling module requires large number of training samples with longer training time, and ends up being specific to the dataset it is trained on, making it non-generalizable to other datasets and other tasks.

Thus, our goal is to present simple, generalizeable and explainable frame sampling modules that do not employ any learning strategy.

2.3. Optical flow

Optical flow is the pattern of apparent motion of image objects between two consecutive frames caused by the movement of object or camera [4]. It is 2D vector field where each vector is a displacement vector showing the movement of points from first frame to second [4].

Optical flow works on two main assumptions.

1. The pixel intensities of an object do not change between consecutive frames [4].

2. Neighbouring pixels have similar motion [4].

Consider a pixel $I(x, y, t)$ in the first frame. It moves by distance (dx,dy) in next frame taken after dt time [4]. So since those pixels are the same and intensity does not change, we can say,

$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$

[4]

Then take Taylor series approximation of right-hand side, remove common terms and divide by dt to get the following equation [4]:

$$f_x u + f_y v + f_t = 0$$

[4] where:

$$f_x = \frac{\partial f}{\partial x}; f_y = \frac{\partial f}{\partial y}$$

[4]

$$u = \frac{dx}{dt}; v = \frac{dy}{dt}$$

[4]

Above equation is called Optical Flow equation [4]. In it, we can find f_x and f_y, they are image gradients [4]. Similarly f_t is the gradient along time [4]. But (u,v) is unknown [4]. We cannot solve this one equation with two unknown variables [4], so several algorithms have been developed to solve this problem.

3. Approach

3.1. Baseline Frame Sampling Strategies

In order to appropriately compare our proposed sampling strategies, we construct simpler sampling strategies to evaluate if our proposed sampling strategies can perform better than these less computationally expensive strategies. In this section, we present three baseline strategies and explain them in detail. In each of the sampling strategies described below, L denotes the length of the original video V, and T denotes the desired length of the final shortened video clip C which is then used as input to the off-the-shelf pre-trained video action recognition model.

3.1.1 Uniform Frame Sampling

In this strategy, a fixed algorithm is used in order to sample the frames used for further processing. First, the original video V of length L is split into T clips of length $\frac{L}{T}$ each. In doing so, we obtain T clips, and the i^{th} clip is denoted by C_i for $i\epsilon\{1, ..., T-1\}$. The clip C_i thus contains frames of indices in the range $[\frac{L}{T}i, \frac{L}{T}(i+1)]$. For each C_i, the first frame is chosen, such that we choose frames at index $\frac{L}{T}i$ for $i\epsilon\{1, ..., T-1\}$. We then concatenated these frames to obtain a single final clip C of length T which is then used for further processing. In this paper, we experiment with T values of $1, 4, 8$, and 16.

3.1.2 Random Frame Sampling

In these strategies, a small number of frames are randomly sampled from the original clip. We experiment with different possible mechanisms of drawing the randomly selected frames, each of which is explained below.

1. **Sequential**: First, a random integer k is drawn uniformly at random in the range $[0, L-1]$. Next, frames with indices in the range $[k, k+T-1]$ are chosen from the original video V and concatenated to obtain the shortened video clip C. In this paper, we only evaluate this strategy for the $T = 16$ case.

2. **Non-Sequential**: First, T random integers are drawn uniformly without replacement from the range $[0, L-1]$. The frames at the indices corresponding to the T random integers drawn are chosen and concatenated together to obtain the shortened video clip C. We experiment with T values of 1, 4, 8 and 16.

3.1.3 Position-based Random Frame Sampling

Similar to random frame sampling, in the following strategies we segregate sections of the original clip into multiple segments, from which we randomly sample frames.

1. **Fourths**: Non-sequential sampling from T sections of the original video V. First, the original video V is split into T clips of equal length, where each clip is denoted as C_i for i in $\{0, ..., T-1\}$. Each clip C_i thus obtained contains $\frac{L}{T}$ frames, and the frames with indices in the range $[\frac{L}{T}i, \frac{L}{T}(i+1)]$ are chosen to be in clip C_i. Thus, a single random frame is drawn from each clip C_i by drawing an integer from the range $[\frac{L}{T}i, \frac{L}{T}(i+1)]$ uniformly at random, and using the frame at that index from the original video V. Each of the frames chosen from each clip C_i is then concatenated together to obtain the final shortened clip C with T frames, as desired. In this paper, we only experiment with $T = 4$.

2. **Mixed**: First frame followed by m frames from each k^{th} section of the original video V. First, the original video V is split into k clips of equal length, where each clip is denoted as C_i for i in $\{0, ..., k-1\}$. Each clip C_i thus obtained contains $\frac{L}{k}$ frames, and the frames with indices in the range $[\frac{L}{k}i, \frac{L}{k}(i+1)]$ are chosen to be in clip C_i. Thus, m random frames are drawn from

each clip C_i by drawing m integers without replacement from the range $[\frac{L}{k}i, \frac{L}{k}(i+1)]$ uniformly at random, and using the frame at that index from the original video V. Along with the first frame of the original video V, each of the m frames chosen from each clip C_i is then concatenated together to obtain the final shortened clip C with $mk+1$ frames, as desired. In this paper, we only experiment with $k = 3$, and $m = 1$, 2, and 5. In the case of $m = 2$, we also add an additional frame obtained by drawing a frame uniformly at random from the original video V, and concatenate this frame with the clip C.

3. **[Beginning, Middle, End] Third**: k frames from the first third, middle third or last third of the original video V. First, the original video V is split into 3 clips of equal length $\frac{L}{3}$. The first, middle and last clip thus obtained are denoted by C_0, C_1 and C_2 respectively. Then, T frames are sampled from either C_0, C_1 or C_2 uniformly at random, and concatenated together to form the final clip C of length T. In this paper, we experiment with T values of $4, 8$, and 16.

3.1.4 Theoretical Best Frame

The goal of this approach is to evaluate how well a model could possibly perform when given only a single frame, i.e. no temporal information at all, assuming that we are somehow able to find a single frame in the video V that allows us to accurately predict the action in the full video V. We find this "theoretical best" performance on a single frame using a procedure described as follows.

1. For every k^{th} frame f_i in the original video V, i.e. stride of k and $i \epsilon \{0, 5, 10, ...\}$, do the following:

   (a) Create a clip C_i by making T copies of f_i and concatenating them together, to obtain a clip of length T.

   (b) Use this clip as input into the pre-trained video action recognition model, and get the model's predictions

   (c) if the model's prediction does not match the ground truth, i.e. it is incorrect, then move on to the next frame. Otherwise, if the model's prediction is correct, i.e. it matches the ground truth, then use this clip C_i for further processing (i.e. $C_i = C$) and stop iterating through the remaining frames.

   In this paper, we experiment only with $k = 5$ and $T = 16$

3.2. Optical Flow-based Sampling Strategies

Previously, we hypothesized that optical flow may provide a signal about which frames may be more salient for action recognition. We test this hypothesis by devising sampling strategies that select frames based on the optical values of each frame. There are many different possible mechanisms for choosing frames based on optical flow. Each of the ones we evaluated in this paper are described below.

First, we begin by calculating the dense optical flow vector between each frame in a given video, which returns $L-1$ dense optical flow vectors. We take the norm of each of the $L-1$ dense optical flow vector to get the optical flow for a specific frame.

1. **Optical Flow (Highest)**: Choose the top T frames that have the highest optical flow norm, and concatenate them together to get clip C used for further processing. In this paper, we experiment with T values of $1, 4, 8$ and 16.

2. **Optical Flow (Lowest)**: Choose the lowest T frames by optical flow norm, and concatenate them together to get clip C used for further processing. In this paper, we experiment with a T value of 16 only.

3. **Optical Flow (Mixed)**: Choose the $T = 8$ highest frames and the $T = 8$ lowest frames by optical flow, and concatenate them together to get clip C used for further processing.

4. **Optical Flow (Smart)**: Details explained in the section below.

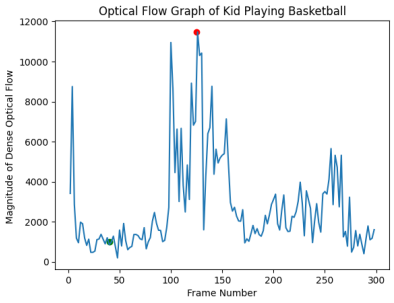
3.2.1 Optical Flow (Smart) Frame Sampling Strategy

(a) Frame 40 (b) Frame 125

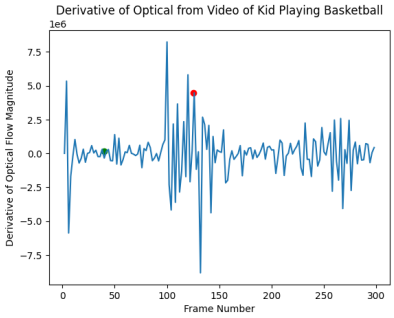
Figure 1. Both frames come from a video of a boy playing basketball. At around frame 125, the person holding the camera drops the phone.

This sampling strategy was developed to help deal with videos that have a high amount of optical flow. One obser-

vation made in the initial parts of this study was that videos that have an excessive amount of movement tend to confuse classifiers that use random, uniform, or position based frame-strategies. This is because there is usually a chunk of the video where the camera is moving around, is dropped, or goes out of focus from the action we are trying to recognize. Often times, when the action is being performed, the optical flow tends to be generally constant and on smaller scales of magnitude than when there is a lot of movement or the camera goes out of focus of the main action. Therefore, we want to use optical flow to identify the regions where the optical flow stays somewhat constant because it is more likely the frames in this region will allow the classifier to accurately predict the action.



(a) Magnitude Optical Flow



(b) Derivative of Magnitude of Optical Flow

Figure 2. Optical Flow and Derivative of Optical Flow Graph for the video of the kid playing basketball.

In Figure 1, we see a video that is labeled with the action "playing basketball". In the beginning of this video, a boy is filmed shooting a basketball at a hoop. In the middle of shooting (from around frames 100-140), the person holding the phone drops the phone and then recovers it at around frame 140. From Figure 2a, we see that the optical flow mirrors this, with the optical flow oscillating drastically from around frames 100-140 and it staying more constant before and afterwards. Figure 2b, which shows the result of taking the derivative of the optical flow, has values close to 0 when the camera is focused on the boy shooting and values close to position of negative 5 when the camera was dropped. To get a good sample of frames to input into our model, we want to choose frames that lie in the regions where the derivative is close to 0.

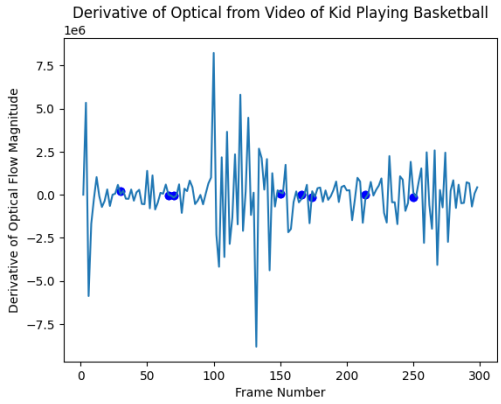


Figure 3. The highlighted points on the derivative of optical flow graph show which frames

Therefore, we define our frame sampling strategy as the following:

1. Compute the derivative of the optical flow for each of the L frames in the video. Take the absolute value of the derivative. Let's call this function $A(f_i)$ for a frame f_i.
2. Split up the video V into clips C_i of equal frame length. For now, let's say that $C_i = \frac{L}{10}$.
3. For each clip C_i, take the average of the absolute value of the derivative of the optical flow. We can represent this as the function $D(C_i) = AVG(A(f_i) \forall f_i \in C_i)$.
4. Choose the T clips with the smallest $D(C_i)$ values.
5. Randomly sample one frame from within each of the T clips and concatenate them to form C.

For this paper, we choose $T = 8$. After running this process, we should get 8 frames from the video that lie in regions with generally constant optical flow. The blue points in Figure 3 show the location on the derivative graph from which the frame sampling algorithm selected frames. We can see that it sampled frames the video that didn't include the parts of the video when the phone fell and was out of focus of the child shooting the basketball.

3.3. Object Detection-based Frame Sampling Strategies

Previously, we hypothesized that the number of objects in a frame may provide a signal about which frames may be more salient for action recognition. We test this hypothesis by devising sampling strategies that select frames based on the number of objects detected in each frame by an off-the-shelf pre-trained model. Each of the strategies we evaluated

in this paper are described below. Since videos may have many frames with the same amount of objects, we tie-break by taking the frames that appear first in chronological order.

1. Choose the top T frames with the largest amount of detected objects, and concatenate them together to get clip C used for further processing. In this paper, we experiment with T values of $1, 4, 8$ and 16.

2. Choose the lowest T frames with the fewest amount of detected objects, and concatenate them together to get clip C used for further processing. In this paper, we experiment with a T value of 16 only.

3. Choose the $T = 8$ highest frames and the $T = 8$ lowest frames in terms of amount of detected objects, and concatenate them together to get clip C used for further processing.

4. Dataset

In order to obtain our test set, we sample 500 examples uniformly at random without replacement from the original validation split of the Kinetics400 dataset [14]. Thus, we evaluate all out proposed frame sampling strategies on this set of videos.

We choose the Kinetics400 dataset [14] since the Kinetics Family is the most widely adopted video action recognition benchmark [36], and thus a good dataset to evaluate our proposed strategies on. Kinetics400 [14] was introduced in 2017, and is the first and most popular dataset published as part of the Kinetics Family, consisting of approximately 240k training and 20k validation videos trimmed to 10 seconds from 400 human action categories [14].

We limit our test set to 500 random examples sampled from the original validation split due to computational and time constraints, since running inference on the full test or validation set would be very computationally expensive, and out of the scope of the time and computational constraints of this paper.

5. Design and Implementation

5.1. VideoMAE [21]: An off-the-shelf action recognition model

This paper uses the base VideoMAE model [21] pretrained for 1600 epochs in a self-supervised way and finetuned in a supervised way on Kinetics-400 [14]. The weights of this off-the-shelf video action recognition model are loaded from HuggingFace [3] from the model checkpoint "MCG-NJU/videomae-base-finetuned-kinetics" [3]. This model checkpoint performs video classification into one of the 400 possible Kinetics-400 [14] labels [14] denoting human actions.

VideoMAE extends Masked Autoencoders (MAE) [12] to video. The architecture of the model is very similar to that of a standard Vision Transformer (ViT) [6], with a decoder on top for predicting pixel values for masked patches [6].

By pre-training the model, it learns an inner representation of videos that can then be used to extract features useful for downstream tasks [2]. For instance, for a dataset of labeled videos, a standard classifier can be trained by adding a linear layer on top of the pre-trained encoder [2]. Typically the linear layer is placed on top of the [CLS] token, as the last hidden state of this token gives a representation of an entire video [2].

5.2. Optical Flow Algorithm

In this paper, we use the an algorithm based on Gunnar Farneback's algorithm [9] as implemented in the OpenCV module "calcOpticalFlowFarneback" [**?**]. This algorithm finds the dense optical flow, which computes the optical flow for all the points in the frame.

5.3. YOLOS [8]: Off-the-shelf object detection model

In this paper, we use the YOLOS tiny-sized model [8], with pre-trained weights loaded from the "hustvl/yolos-tiny" checkpoint from HuggingFace [**?**].

YOLOS [8] is fine-tuned on the COCO 2017 object detection [17] containing 118k annotated images [17]. It can perform object detection, and is able to detect 80 classes of objects from the COCO dataset [17].

5.4. Model Pipeline and Experiment Setup

To evaluate which frame sampling strategies are the most effective, we run all the frame sampling strategies defined in section 3 with their respective parameters and obtain 16 frames to input into the model. If our frame sampling strategy is supposed to return less than 16 unique frames, we repeat the unique frames so that we can input 16 frames into the model. For example, if we sample one unique frame, we will repeat that frame sixteen times for input into the model. The sampled frames are fed into a pretrained VideoMAE (finetuned on Kinetics400) classifier to perform action recognition on a test set of 500 randomly sampled videos from the Kinetics400 dataset. We report the accuracy of the classifier on those 500 videos in our results.

5.5. Hypotheses

Based off our proposed sampling strategies, we hypothesize the following:

1. Both object detection based sampling and optical flow based sampling will yield better results than our baseline sampling strategies. Because both object detection and optical flow consider video information such

as motion and presence of objects, frames selected by these strategies will likely be of higher "value" compared to randomly sampled frames, and thus allow the model to improve its inference accuracy.

2. Sampling strategies which sample more frames will yield higher accuracy and perform better than sampling strategies which sample fewer frames. However, single frame sampling strategies will still yield better than random results, due to the fact that a single frame will likely be a good representation of the entire video, and contain overlapping information with other frames.

6. Results, Analysis and Discussion

6.1. Quantitative Analysis

6.1.1 Random chance of success

The Kinetics400 dataset [14] contains 400 classes of actions, so $\frac{1}{400} = 0.0025$ is the random chance accuracy. This indicates that all of our proposed approaches capture some significant enough signal that allows them to select the most salient/discriminative frames for action recognition. This makes sense since even a single frame would contain some amount of information that would allow for the model to rule out completely unrelated classes, and thus narrow down the possible action classes the frame could possibly represent considerably. For instance, a picture of a person in a field outdoors with a ball completely rules out the possibility that the action is cooking, since it is impossible that someone outdoors with a ball could possibly be cooking, especially in the absence of any cooking material.

6.1.2 Comparing Clip Sampling Strategies

See Table 1 for a complete list of accuracies for all sampling methods. We found that all of our sampling methods resulted in a classification accuracies above 65%, when using 16 frames.

Our baseline strategies: Uniform sampling performs the best across all of our frame sampling strategies (with the exception of the theoretical best, of course). The non-deterministic random and position-based sampling strategies performed slightly worse than uniform sampling, potentially due to uncertainty and randomness affecting the quality of the selected frames.

Among the random sampling strategies, it is clear that sequentially sampling frames leads to a significant drop in performance as compared to non-sequentially sampling frames. For videos that have a high temporal variability, randomly sampling non-sequentially gives frames that encode greater temporal information and visual cues, which may be the reason for the improvement in performance here. We hypothesize that this is also the reason why uniform sampling outperforms both random sampling and position-based sampling.

We found that our proposed optical flow and object detection strategies did not contain any significant improvement over our baselines, with the maximum accuracy from each category being roughly similar and within a 10 video margin. Within the optical flow category, we see that our smart frame sampling strategy performs better than the other frame sampling strategies, with the mixed frame-sampling strategy being a close second. One observation that we make in our analysis is that the optical flow (highest and lowest) models have a smaller variance between their sampled frame position numbers compared to random and position based sampling methods. This indicates that sampling frames by optical flow causes us to sample frames that are temporally close to each other which may prove problematic for videos that have high temporal variance. The variance of the frame positions with the smart and mixed optical flow sampling model are higher than that of the other optical flow sampling strategies which might be the reason for it's improvement in performance.

Our object detection based sampling strategy performed more poorly compared to optical flow based sampling in every equivalent experiment, and performed slightly worse on our baseline strategies. The highest accuracy for object detection sampling was 76.8%, which is less than the 78.2% accuracy of smart optical flow sampling and 77.6% and 78.8% accuracies of the position based and uniform baseline strategies.

Additionally, we found that the theoretical best accuracy for the video action recognition model when using a single frame with no temporal information was 85.9%, which is the highest accuracy of all our results. This means that some videos contained frames which we did not accurately sample, but would have contributed to a correct classification.

When considering the effects of sampling fewer frames in Table 2, our baselines even out performed our optical flow and object detection based sampling strategies for the 16 and 4 unique frame experiments.

These results reject our hypothesis 1, and show that there is no significant difference in accuracy between our proposed object detection and optical flow based sampling methods.

6.1.3 Comparing Number of Unique Frames

When comparing our sampling strategies which sample different amounts of unique frames per video, we found that for the majority of cases, sampling more frames sampled always yielded better accuracies.

For sampling strategies which only sampled one frame,

	Sampling Method	Unique Frames	Accuracy
Random	**Sequential**	16	0.672
	Non-Sequential	16	**0.766**
	Uniform	16	**0.788**
Position Based	**Fourths**	4	0.664
	Mixed	16	**0.776**
	Beginning Third	16	0.736
	Middle Third	16	0.764
	End Third	16	0.746
Optical Flow	**Highest**	16	0.770
	Lowest	16	0.734
	Mixed	16	0.776
	Smart	8	**0.782**
Object Detection	**Highest**	16	0.764
	Lowest	16	0.704
	Mixed	16	**0.768**
	Theoretical Best	1	**0.859**

Table 1. Table of classification accuracies for each sampling method. Each sampling method involved sampling unique frames from a video. Our baseline results from Random Sequential, Random Non-Sequential, Uniform, and Position-based sampling are compared to our proposed Optical Flow and Object Detection based sampling methods.

Sampling Method	**Number of Unique Frames**			
	16	**8**	**4**	**1**
Random Non-Sequential	0.766	0.730	0.662	0.496
Uniform	**0.788**	0.700	0.640	0.476
Position (Mixed)	0.776	0.718	**0.672**	-
Position (Beginning Third)	0.736	0.668	0.612	0.490
Position (Middle Third)	0.764	0.702	0.662	0.540
Position (End Third)	0.746	0.676	0.604	0.514
Optical Flow (Highest)	0.770	0.652	0.594	0.488
Optical Flow (Smart)	-	**0.782**	-	-
Object Detection (Highest)	0.764	0.700	0.650	**0.554**

Table 2. Table of classification accuracies comparing our baseline sampling methods to our Optical Flow and Object Detection based sampling methods on different sizes of video clips (16, 8, 4, and 1 unique frames).

we found that object detection yielded the highest accuracy at 55.4%, while the other sampling strategies performed slightly worse, with accuracies around 50%. This indicates that for single frame sampling, object detection seemed to be the most effective. For sampling strategies which sampled four frames, the best performing model was the smart optical-flow model and for the sampling strategy which sampled eight frames, the random non-sequential model did the best. These results are in line with our hypothesis 2, with single frame classification accuracies being significantly higher than random guessing.

One question we tried to answer in this paper was whether it is possible to find a method of representing the same temporal information in a video with fewer frames. For example, is it possible to classify the correct action in a video by just using one frame? The theoretical best model shows that for 85.9% of the videos in this dataset, it is possible for us to correctly classify the right action with just one frame of the video. Unfortunately, none of of our clip-sampling methods was able to achieve the same kind of accuracy with the highest accuracy being only 55.4%. However, we can see that it is theoretically possible to see a classifier that uses fewer than 16 frames having acceptable model performance, which increases scalability and efficiency for action recognition models if such a frame-sampling algorithm is found. Notably, our **smart optical-flow** model does outperform the the majority of our other models with only 8 frames instead of 16. This shows that it would actually be more computationally efficient for VideoMAE to implement our smart optical flow frame-sampling strategy and only require 8 frames to classify with the same level of accuracy that their current model does with 16

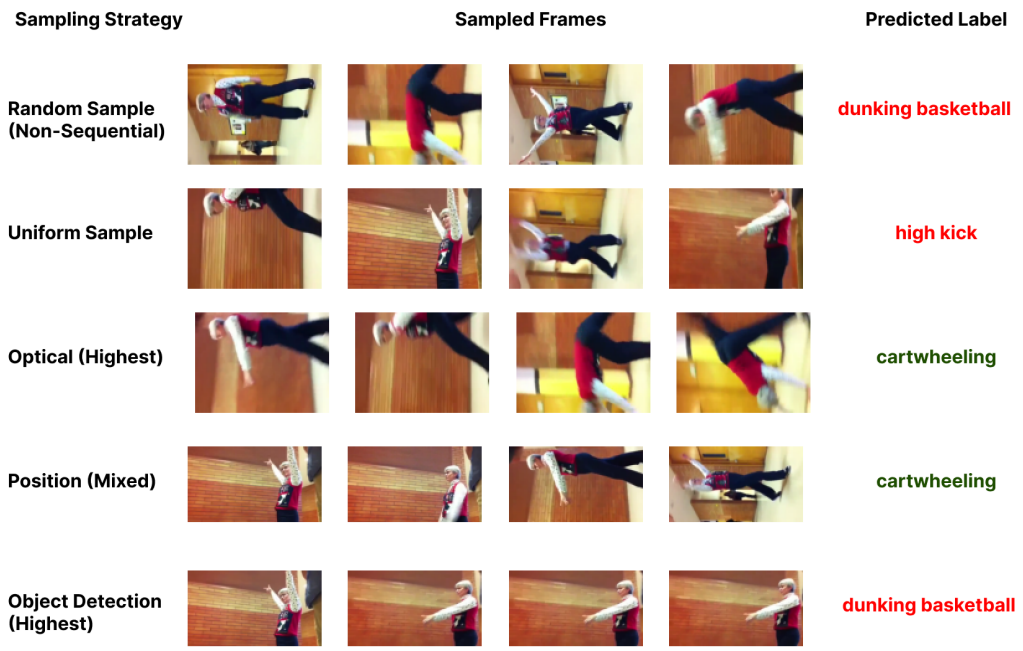


Figure 4. **Qualitative Analysis.** This figure shows the results of running 5 different 4-frame clip sampling methods on a video of a woman cartwheeling. We see that only optical and position are able to extract frames to allow the model to predict the correct label.

frames.

6.2. Qualitative Analysis

In Figure 4, we have an example of four frames extracted from a video of a woman cartwheeling by different frame sampling strategies that we explore in the paper as well as the label that was predicted from the extracted frames. These clip sampling strategies only extract 4 frames from the video for classification. There are a couple of insights that we can gain from this figure. First of all, in the video, we know that the woman only cartwheels for a part of the video, which means that extracting the 4 frames from the only part of the video in which the woman is cartwheeling is important for classification.

Right off the bat, we see that objection detection, uniform sampling, and random sampling predict the wrong label. For random sampling, it gets two frames of the woman standing at the end of the cartwheel which isn't helpful for the model to understand that the woman is cartwheel. Uniform sampling isn't able to get the woman mid-cartwheel, completely passing over the part of the video where she is cartwheeling (even a human might not be able to tell that she was cartwheeling if given those frames). Object detection simply outputs frames from the beginning of the video because it doesn't detect the woman mid-cartwheel as an object.

However, both the optical flow model and the position (mixed) model are able to predict that the woman is cartwheeling. The optical flow model grabs the frames where the woman is mid-cartwheel as well as the frames right before she cartwheels which is enough for the model to properly classify her action. This helps us understand how using optical flow for frame sampling may be beneficial for certain action recognition tasks. In this video, using optical flow to sample frames worked well because the predicted action class involved a lot of activity and was concentrated in a subsection of the video, so methods like uniform and random sampling had a hard time extracting appropriate frames.

6.2.1 Action Classes that Require More Frames to Classify

Another point of analysis in this paper was trying understanding which action classes require more frames to correctly classify. To do this, we found the classes that were most misclassified by the 1-frame, 4-frame, and 8-frame random-sampling (non-sequential) clip-sampling method as compared to the 16-frame random-sampling (non-sequential) method. The top 10 classes that were misclassified because there were less frames extracted are: {blowing nose, playing basketball, climbing a rope, sharpening pencil, peeling apples, skiing slalom, cartwheeling, strumming guitar, tap dancing, swimming breast stroke}. This result is particularly curious because not all of these

classes are high-action as we had initially hypothesized. However, when digging into some of the videos that were misclassified within these action classes, we are able to see that these misclassifications are a result of a large amount of temporal variance in the video that smaller number of frames aren't able to capture.

6.2.2 Strengths and Weaknesses of Our Clip-Sampling Strategies

- **Random and Uniform Frame-Sampling Techniques:** Random and uniform frame-sampling techniques generally tend to perform well because they tend to grab frames from multiple parts of the video. Obviously, they underperform when trying to extract 1 frame but when extracting multiple frames (16,8, etc), they tend to extract frames that give a holistic understanding of the video, especially if the video has high temporal variance. However, random, uniform, and position based sampling methods struggle if the temporal activity all resides in a small subset of the video (for example if the action occurs only towards the end of the video).

- **Optical Flow Frame-Sampling Techniques:** Optical Flow frame sampling techniques tend to outperform random, uniform, and position based sampling methods when the action is the video involves a lot of motion and person or thing doing the action takes up a large portion of the video. Optical flow frame-sampling methods fail oftentimes when they attempt to predict action classes that have less motion associated with them (i.e. peeling apples, sharpening pencil vs cartwheeling, tap dancing). They also fail often times when the action happening takes up a small part of the frame, which means that the optical flow values calculated are lower than they would be if we zoomed in on the video.

- **Object Detection Frame-Sampling Techniques:** Object detection techniques succeed when multiple actors take place in doing an action or where multiple objects are involved in the action being performed. The ability of an object detection frame-sampling mechanism to extract the correct frames is severely limited if the action being done doesn't involve any discernable objects or if the video being classified involves additional objects moving in and out of the frame that are not associated with the action we are trying to classify. In addition, running object detection models on frames takes needed computation and causes inference to take longer.

7. Conclusion and Future Work

In this paper, we investigate two possible smart sampling strategies along with more naive baseline strategies to evaluate possible sampling strategies for more resource-efficient video action recognition. We find that although our hypotheses hold true to some extent, i.e. there is some additional signal provided by the optical flow norm and number of objects in the frame about whether the frame is salient for action recognition, our proposed approaches do not perform significantly better than some of the less computationally expensive baselines introduced later in the paper. However, one important insight we notice is that our 8-frame smart optical flow frame-sampling method is able to achieve the same accuracy as other frame sampling strategies that extract 16 frames which means that it is possible to use 8 frames rather than 16 frames to perform action recognition with high accuracy, which can lead to improvements in efficiency and scalability for action recognition platforms. We also are able to evaluate and do an in-depth analysis on the strengths and weaknesses of these frame-sampling strategies on different types of videos. [future work suggestions] are all promising areas for future work.

References

[1] An Introduction to Video Understanding: Capabilities and Applications — blog.fastforwardlabs.com. https://blog.fastforwardlabs.com/2021/12/14/an-introduction-to-video-understanding-capabilities-and-applications.html. [Accessed 07-May-2023]. 1

[2] MCG-NJU/videomae-large-finetuned-kinetics · Hugging Face — huggingface.co. https://huggingface.co/MCG-NJU/videomae-large-finetuned-kinetics. [Accessed 10-May-2023]. 7

[3] Models - Hugging Face — huggingface.co. https://huggingface.co/models. [Accessed 10-May-2023]. 7

[4] OpenCV: Optical Flow — docs.opencv.org. https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html. [Accessed 07-May-2023]. 2, 3, 4

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset, 2018. 3

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 7

[7] Hehe Fan, Zhongwen Xu, Linchao Zhu, Chenggang Yan, Jianjun Ge, and Yi Yang. Watching a small portion could be as good as watching all: Towards efficient video classification. In *Proceedings of the Twenty-Seventh International*

Joint Conference on Artificial Intelligence, IJCAI-18, pages 705–711. International Joint Conferences on Artificial Intelligence Organization, 7 2018. 2

[8] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu. You only look at one sequence: Rethinking transformer in vision through object detection, 2021. 7

[9] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. volume 2749, pages 363–370, 06 2003. 7

[10] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition, 2020. 3

[11] Ruohan Gao, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani. Listen to look: Action recognition by previewing audio, 2020. 3

[12] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. 7

[13] Amlan Kar, Nishant Rai, Karan Sikka, and Gaurav Sharma. Adascan: Adaptive scan pooling in deep convolutional neural networks for human action recognition in videos, 2017. 3

[14] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017. 7, 8

[15] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition, 2019. 3

[16] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling, 2021. 3

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. 7

[18] Washington L. S. Ramos, Michel M. Silva, Mario F. M. Campos, and Erickson R. Nascimento. Fast-forward video based on semantic extraction. In *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, sep 2016. 3

[19] Zheng Shou, Xudong Lin, Yannis Kalantidis, Laura Sevilla-Lara, Marcus Rohrbach, Shih-Fu Chang, and Zhicheng Yan. Dmc-net: Generating discriminative motion cues for fast compressed video action recognition, 2019. 2

[20] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos, 2014. 1

[21] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training, 2022. 7

[22] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks, 2015. 3

[23] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition, 2018. 1

[24] Oskar Triebe, Nikolay Laptev, and Ram Rajagopal. Ar-net: A simple auto-regressive neural network for time-series, 2019. 3

[25] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition, 2016. 1

[26] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos, 2017. 3

[27] Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R. Manmatha, Alexander J. Smola, and Philipp Krähenbühl. Compressed video action recognition, 2018. 2

[28] Wenhao Wu, Dongliang He, Xiao Tan, Shifeng Chen, and Shilei Wen. Multi-agent reinforcement learning based frame sampling for effective untrimmed video recognition, 2019. 2, 3

[29] Zuxuan Wu, Caiming Xiong, Yu-Gang Jiang, and Larry S. Davis. Liteeval: A coarse-to-fine framework for resource efficient video recognition, 2019. 3

[30] Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S. Davis. Adaframe: Adaptive frame selection for fast video recognition, 2019. 2, 3

[31] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos, 2017. 3

[32] Bowen Zhang, Limin Wang, Zhe Wang, Yu Qiao, and Hanli Wang. Real-time action recognition with enhanced motion vector cnns, 2016. 2

[33] Yin-Dong Zheng, Zhaoyang Liu, Tong Lu, and Limin Wang. Dynamic sampling networks for efficient action recognition in videos. *IEEE Transactions on Image Processing*, 29:7970–7983, 2020. 3

[34] Yuan Zhi, Zhan Tong, Limin Wang, and Gangshan Wu. MGSampler: An explainable sampling strategy for video action recognition, 2021. 1, 2, 3

[35] Wangjiang Zhu, Jie Hu, Gang Sun, Xudong Cao, and Yu Qiao. A key volume mining deep framework for action recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1991–1999, 2016. 3

[36] Yi Zhu, Xinyu Li, Chunhui Liu, Mohammadreza Zolfaghari, Yuanjun Xiong, Chongruo Wu, Zhi Zhang, Joseph Tighe, R. Manmatha, and Mu Li. A comprehensive study of deep video action recognition, 2020. 1, 2, 3, 7