data

MDPI

Article

Unsupervised Few Shot Key Frame Extraction for Cow Teat Videos

Youshan Zhang [1,*], Matthias Wieland [2] and Parminder S. Basran [1]

[1] Department of Clinical Sciences, College of Veterinary Medicine, Cornell University, Ithaca, NY 14853, USA; psb92@cornell.edu

[2] Department of Population Medicine and Diagnostic Sciences, College of Veterinary Medicine, Cornell University, Ithaca, NY 14853, USA; mjw248@cornell.edu

* Correspondence: yz945@cornell.edu

Abstract: A novel method of monitoring the health of dairy cows in large-scale dairy farms is proposed via image-based analysis of cows on rotary-based milking platforms, where deep learning is used to classify the extent of teat-end hyperkeratosis. The videos can be analyzed to segment the teats for feature analysis, which can then be used to assess the risk of infections and other diseases. This analysis can be performed more efficiently by using the key frames of each cow as they pass through the image frame. Extracting key frames from these videos would greatly simplify this analysis, but there are several challenges. First, data collection in the farm setting is harsh, resulting in unpredictable temporal key frame positions; empty, obfuscated, or shifted images of the cow's teats; frequently empty stalls due to challenges with herding cows into the parlor; and regular interruptions and reversals in the direction of the parlor. Second, supervised learning requires expensive and time-consuming human annotation of key frames, which is impractical in large commercial dairy farms housing thousands of cows. Unsupervised learning methods rely on large frame differences and often suffer low performance. In this paper, we propose a novel unsupervised few-shot learning model which extracts key frames from large (∼21,000 frames) video streams. Using a simple L1 distance metric that combines both image and deep features between each unlabeled frame and a few (32) labeled key frames, a key frame selection mechanism, and a quality check process, key frames can be extracted with sufficient accuracy (F score 63.6%) and timeliness (<10 min per 21,000 frames) for commercial dairy farm setting demands.

Keywords: key frame extraction; dairy cows; unsupervised few shot learning

check for updates

Citation: Zhang, Y.; Wieland, M.; Basran, P.S. Unsupervised Few Shot Key Frame Extraction for Cow Teat Videos. *Data* **2022**, *7*, 68. https://doi.org/10.3390/data7050068

Academic Editors: S. Ejaz Ahmed, Shuangge Steven Ma and Peter X.K. Song

Received: 22 April 2022
Accepted: 18 May 2022
Published: 23 May 2022

1. Introduction

Monitoring the dairy cows' health is critical in ensuring quality milk production. In the commercial dairy farm setting, monitoring the health of thousands of cows is a time-consuming and expensive task. During the milking process, cows are moved toward large parlors for machine milking as shown in Figure 1. These systems consist of a large and slowly rotating set of stalls, where a cow is guided into a stall, a milking unit is manually attached to the cow's teats, machine milking commences via vacuum, and the milking unit automatically detaches and retracts from the teats. Thereafter, the cow exits the rotating parlor.

Within a milking session, the opportunity for a veterinarian to assess the health of the dairy cows' teats is limited to immediately before or after the milking unit attaches or detaches from the teats. Mastitis, or bacterial infections of the udders and/or teats, poses one of the greatest health concerns for dairy cows. The risk of mastitis is increased with changes in the callosity (hyperkeratosis) of the teat end, and this can be assessed via manual inspection. While it is possible to assess the extent of hyperkeratosis in a large proportion of the herd during a milking session, the total time available for the veterinarian to conduct this assessment is limited due to the finite amount of time that the cow is in the stall (typically tens of seconds). It is thus impractical to conduct health assessments of the entire herd in this manner, and industry standards suggest evaluating 20% (or greater) of the herd [1].

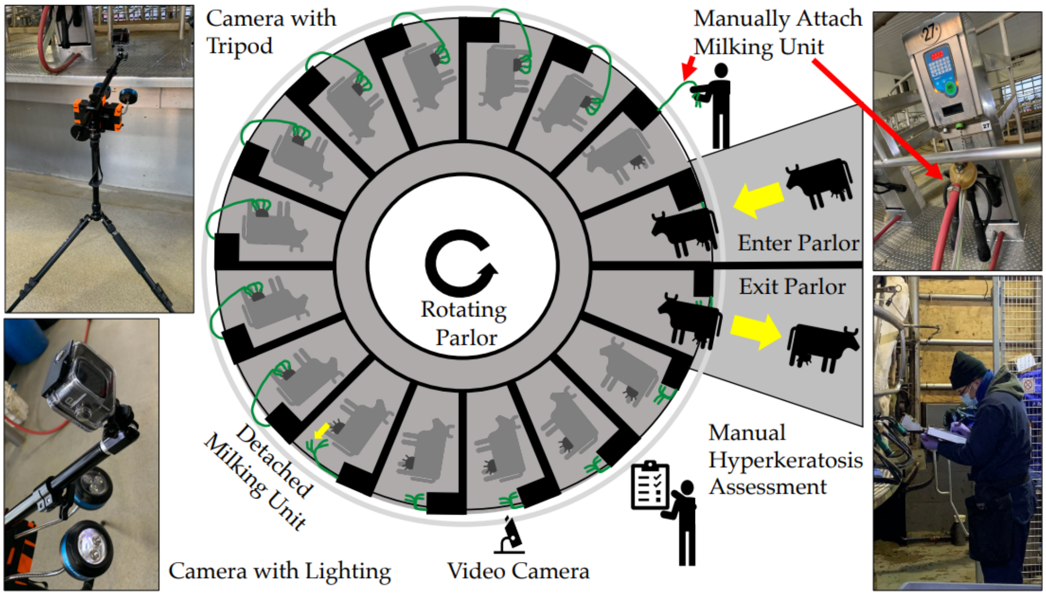


Figure 1. The milking machine of large dairy farm. Videos are recorded when the parlor is rotating. The two cameras shown on the left correspond to the video camera (below the rotary parlor).

Recently, we proposed a digital framework for evaluating the extent of hyperkeratosis by using a digital camera and software [2]. This approach allows remote assessment of the entire population of cows that enter the parlor. A digital approach also permits the opportunity for several experts to conduct assessments of hyperkeratosis independently and mitigate the influence of inter-rater variability. We have also shown that it is feasible to use deep learning to classify the extent of hyperkeratosis [3]. These innovations permitted the opportunity to explore whether such health assessments can be conducted remotely using video-based imaging systems. Later, we proposed a separable confident transductive learning [4] model to minimize the difference between training and test datasets, and we improved the hyperkeratosis recognition accuracy from 61.8 to 77.6%.

While a video-based analysis might seem like a simple extension of this work, analyzing the entire video frame by frame is inefficient since only a small number of frames contain useful diagnostic information. Many vision-based tasks (classification, segmentation) can be performed more efficiently using key frames (KFs) instead of the full video, thus one option is to select KFs from these cow teat videos for analysis. Most existing key frame extraction (KFE) methods use supervised or unsupervised learning. Supervised learning requires the manual labeling of KFs from large-scale training data to train a model. In the dairy farm setting, it is not practical nor economical to manually label all video images; thus, unsupervised or semi-supervised learning models are preferred. Unsupervised learning models for detecting KFs rely on significant changes between image frames. The cognitive goal of our problem is to extract key teat frames from video sequences in which changes in objects between frames are less obvious. The utilitarian goal is to efficiently and accurately extract key frames with only a few key frames. Therefore, existing supervised (require massive labels) and unsupervised (require significant frame changes) methods are ineffective in our problem.

We propose a modified few-shot learning approach and leverage knowledge from several (N = 32) support KFs and then identify KFs in unlabeled video image frames (Figure 2). Figure 3 shows 6 of the 32 KFs used in this study. This paper provides three specific contributions:

- **The CowTeatVideo Benchmark**. We provide a new, publicly available dataset consisting of dairy cow teat videos for key frame extraction. This is a published dataset of dairy cow teat videos that can be used for the testing and evaluation of different KFE models.

- **Few-shot generalized learning**. We run few-shot learning without a base training dataset and unlabeled query datasets (cow teat videos). The key frames are detected using the distance between unlabeled query datasets and support key frame images.
- **UFSKFE model**. We describe a novel unsupervised few-shot learning key frame extraction (UFSKFE) model for our problem. We combine the L1 distance of raw RGB images and extracted deep features to form a robust fusion distance. After selecting key frame candidates, we further propose a quality check process to remove noisy key frames.

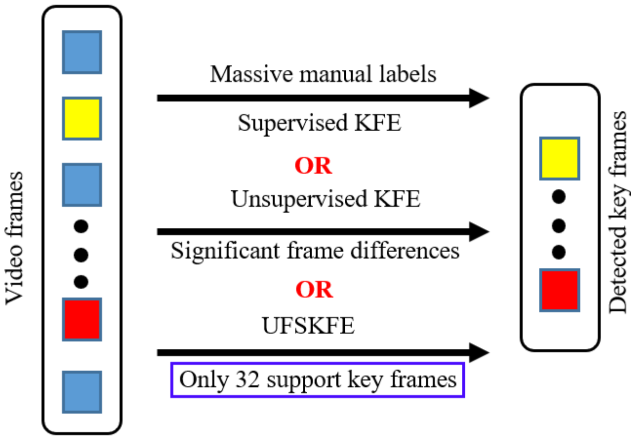


Figure 2. Differences between existing supervised key frame extraction (KFE), supervised KFE, and our proposed unsupervised few-shot KFE (UFSKFE) model.

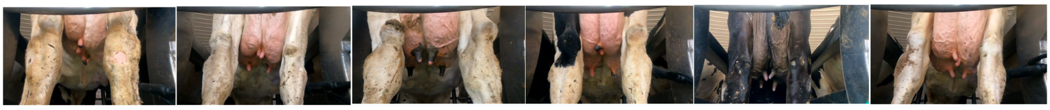


Figure 3. Six sample key frames (KFs) in the cow teat video. These KFs should provide a clean, unambiguous, and high-resolution image of the dairy cow teats for clinical diagnosis, suppress similar frames, and be diverse enough to reduce redundancy.

2. Related Work

Extracting correct KFs has been a long-standing problem with many applications, such as managing, storing, transmitting, and retrieving video data. Both traditional and deep learning-based methods have been explored.

2.1. Traditional Methods

Traditional KFE models can be divided into two categories: unsupervised learning and supervised learning. Unsupervised KFE often relies on computing the relevance, diversity and representations using extracted traditional features using optical flow [5,6], SIFT [7,8] and SURF features [9,10]. The clustering approach is one representative unsupervised KFE method [11]. Mendi and Bayrak [12] developed a dynamic KFE method through three steps: color histogram differences, self-similarity modeling and unsupervised k-means clustering. Priya and Dominic [13] utilized inter-cluster similarity analysis to extract KFs. Vázquez-Martín and Bandera [14] computed similarity by building an auxiliary graph of frame features and then applied spectral clustering to extract KFs. Later, Ioannidis et al. [15] extracted KFs via applying spectral clustering to a composite similarity matrix that was computed using weighting sum of all similarity matrices of video frames. Supervised KFE models rely on human annotated data to train a machine learning model and generate KFs from the test videos. Ghosh et al. [16] and Gygli et al. [17] treated the process of extracting KFs as a regression scoring problem, where a higher score is selected as a KF. Yao et al. [18]

proposed a multifeature fusion method (which can capture complicated and changeable dancer motions) to extract KFs from dance videos.

2.2. Deep Learning Models

Recently, deep learning approaches have attracted interest in KFE. Both supervised and unsupervised deep learning models have been proposed to boost the performance of KFE from videos. Supervised deep KFE models usually estimate a frame's importance via deep neural networks with the aid of ground truth KFs. Zhang et al. [19] first applied long short-term memory (LSTM) units to model variable-range temporal dependency among video frames, and they predicted the frame's importance via multi-layer perceptron. Later, Zhao et el. [20] proposed a two-layer LSTM to estimate the key fragments of a video. They further developed a tensor-train embedding layer in a hierarchical architecture of recurrent neural networks to model the long temporal dependency among video frames [21]. Based on [19], Casas and Koblents introduced an attention mechanism to estimate the frame's importance and select the video KFs. Fajtl et al. [22] utilized self-attention with a two-layer fully connected network to predict the frame's importance score. Li et al. [23] developed a global diverse attention mechanism based on a pairwise similarity matrix that contains diverse attention weights. These weights can further transform into frame importance scores. Jian et al. [24] extracted the KFs of sports videos, considering the neighboring probability difference of frames, and these probabilities were estimated from a CNN on extracted region of interest areas. Yuan et al. [25] introduced a global motion model to extract candidate KFs; spatial–temporal consistency and hierarchical clustering were used to extract KFs.

There are also several unsupervised deep learning models for KFE. Yuan et al. [26] introduced a bidirectional LSTM model to automatically extract KFs. Mahasseni et al. [27] applied the generative adversarial networks (GAN) in KFE. They employed an LSTM as a frame selector and confused the discriminator (which aims to distinguish original video and reconstructed video). Yuan et al. [28] utilized bidirectional LSTM as a frame selector to model the temporal dependency among frames, and KFs were evaluated by two GANs. Yan et al. [29] proposed an automatic self-supervised learning model to detect KFs in videos. They proposed to generate pseudo labels for each frame with optical flow and RGB image features. Li [30] proposed an end-to-end network embedding for unsupervised KFE for person re-identification. They designed a KFE module by training a CNN with pseudo labels generated by hierarchical clustering. Recently, Elahi and Yang [31] proposed an online learnable module for KFE, and the extracted KFs were used for recognizing action with deep learning-based classification models.

Our goal is to devise an effective strategy to extract KFs that contain a clear, unambiguous, and high-resolution image of the dairy cow teats for clinical diagnosis. Unsupervised learning models rely on the sharp differences between consecutive frames to determine the KFs, but this is not the case in our problem. Unsupervised clustering models can lead to low performance in our situation since KFs are similar to each other and may be easily be assigned to the same class (see sample KF images in Figure 3).

Few-shot learning aims to accomplish a learning task by using very few training examples, which typically recognize the different categories of images in the query dataset given a base training dataset and a support dataset [32–34]. Oreshkin et al. [35] trained a normal global classifier on the base dataset to form an auxiliary task, which can co-train the few-shot classifier and create a regularization effect. Gidaris et al. [36] combined self-supervision with few-shot learning, which can learn rich and transferable visual representations with few annotated samples. Hong et al. [37] utilized reinforcement learning for training an attention agent to generate discriminative representation in few-shot learning. Wei and Mahmood [38] optimized few-shot learning tasks by generating new samples using variational autoencoders on face recognition. However, current few-shot models are mostly supervised and rely on labeled examples. Current attempts of unsupervised few-shot learning [39,40] are not suitable in our problem. Only a few KFs (support dataset) and unlabeled cow teat videos are provided for the learning.

3. Methodology

3.1. *Motivation*

Given the unique nature of our dataset and problem, we propose to apply few-shot learning in an unsupervised manner for KFE. We then design a framework that takes the knowledge from the few support KF images to find its nearby neighbors using both raw RGB images and pre-trained deep features distances as shown in Figure 4.

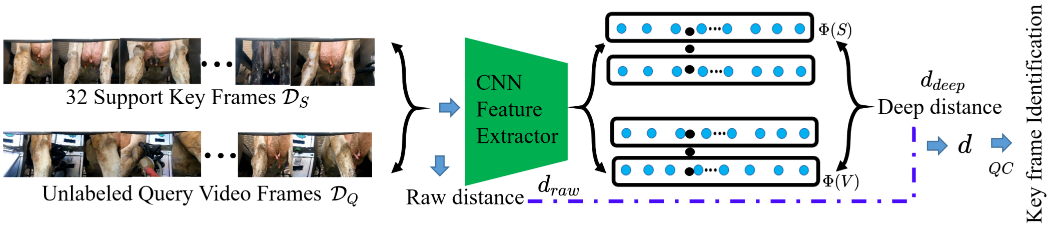


Figure 4. The scheme of our proposed unsupervised few-shot key frame extraction (UFSKFE) model. We first calculate the raw distance d_{raw} between each video frame image and few support key frame (KF) images. Secondly, we employ a pre-trained CNN (ResNet-101) to extract deep features for video frame images $\Phi(V)$ and 32 support key frames ($\Phi(S)$) and then calculate the deep distance d_{deep}. Lastly, we perform a quality check (QC) to select KFs for each video with a smaller fusion distance (d).

3.2. *Preliminaries*

3.2.1. Key Frame Extraction

Given a video $V = \{v_i\}_{i=1}^{n_v}$, where v_i is the i-th frame image and n_v is the number of frames in video V, the goal of video KFE is to fetch the KF numbers $\mathcal{Y}$:

$$\mathcal{Y} = \mathcal{S}(V), \tag{1}$$

where $\mathcal{Y} = \{y_j\}_{j=1}^{n_y}$ (n_y is the number of predicted KFs, and $n_y << n_v$) and $\mathcal{S}$ is an automatic KF selection function. In supervised KFE, the KF numbers $F = \{f_i\}_{i=1}^{n_f}$ of video V, or importance of each frame image, are provided, where n_f is the number of KFs and typically $n_f << n_v$. We aim to minimize the error between $\mathcal{Y}$ and F during the training and generalize the trained model for new video data. In unsupervised KFE, no KFs are known (i.e., $F = \varnothing$). It aims to predict $\mathcal{Y}$ that can best describe the content of a video V.

3.2.2. Few Shot Learning

In supervised few-shot learning, we have a labeled base training dataset $\mathcal{D}_{base} = (\mathcal{X}, \mathcal{Z}) = \{x_i, z_i\}_{i=1}^{n_d}$ that contains n_d labeled training images from A base classes, i.e., $z_i \in \{1, 2, \cdots, A\}$. In addition, we are given a support dataset $\mathcal{D}_S$ of labeled images from C novel classes, and each class has K examples. The goal of few-shot learning is to train a model that can accurately recognize the C novel classes in another query dataset $\mathcal{D}_Q$. This learning paradigm is called C-way K-shot learning. In unsupervised few-shot learning, there are no labels for the base training dataset, i.e., $\mathcal{D}_{base} = \mathcal{X} = \{x_i\}_{i=1}^{n_d}$. In our KFE problem, the base training dataset is also unavailable i.e., $\mathcal{D}_{base} = \varnothing$. We treat the full video as the query dataset, and it has no labels. In the next section, we discuss how we can construct tasks in unsupervised KFE with few-shot learning.

3.3. *Unsupervised Few-Shot KFE*

In traditional unsupervised KFE, poor performance is often the result of no labeled KFs. In our videos, there are no distinctive changes between frames, like in sports videos. To improve the learning of these KFs, we start with a few KFs (i.e., a support dataset $\mathcal{D}_S$ exists). Since we only have one class (KFs) and K KFs (K images, $K = 32$ in our case), our problem can be treated as a one-way 32-shot problem, or a few-shot learning perspective. However, the aforementioned base training dataset is not provided. Furthermore, the query dataset is our unlabeled cow teat video ($\mathcal{D}_Q = V$). A key question then is how to obtain key frames from each cow in all unlabeled videos with only a few prior KFs. Inspired by few-shot learning, we consider measuring the distance between each video frame image and support KFs.

3.3.1. Raw Distance Representation

To select KFs from the unlabeled videos, we propose to calculate the distance between support KF images $\mathcal{D}_S = \{s_k\}_{k=1}^{K=32}$ and each frame image of a video. Frames with the lowest distances could be potential KFs. First, we calculate a distance based on each raw frame image and support KF image via the distance matrix $M_{raw} \in \mathbb{R}^{n_v \times K}$ in Equation (2), which represents the L1 difference between each raw video frame image and K support KF raw images. An element in the distance matrix is defined as

$$M_{raw}^{ik} = |s_k - v_i|_1, \tag{2}$$

where $|\cdot|_1$ is the L1 norm of the difference between one support KF image and one video frame ($k \in \{1, \cdots, K\}$ and $i \in \{1, \cdots, n_v\}$), $|s_k - v_i|_1 \in \mathbb{R}^{1\times 1}$, and hence, $M_{raw} \in \mathbb{R}^{n_v \times K}$. We then define the raw distance as

$$d_{raw} = \min_r M_{raw}, \tag{3}$$

where $\min_r$ returns the minimum number of each row in the matrix M_{raw}. For each frame v_i, its associated raw distance is $d_{raw}^i = \min\{M_{raw}^{ik}\}_k^K \in \mathbb{R}^{1\times 1}$ and denotes the distance to one of the closest support KF images. Since a video contains many images of a cow—and many cows—several KFs to compare against an analyzed image are necessary. We aim to have a diverse set of support key frame KF s_k from which at least one image closely resembles the current frame. For all frames in any video V, we can calculate the raw distance $d_{raw} \in \mathbb{R}^{n_v \times 1}$. Note, however, that the raw distance is computed using original images and might not capture all of the important features in a key frame. So, we describe how we extract deep features from both the video frame images and support KF images, and then calculate a deep feature distance, as described in more detail in the next section.

3.3.2. Deep Distance Representation

There is no deep model for cow teat video classification or segmentation; thus, our approach is to extract deep features from a pre-trained ImageNet model. Let Φ represent feature extraction from a pre-trained ImageNet model. Similar to the raw distance matrix in Equation (2), an element in a deep distance matrix is denoted as

$$M_{deep}^{ik} = |\Phi(s_k) - \Phi(v_i)|_1, \tag{4}$$

where $\Phi(\cdot) \longrightarrow \mathbb{R}^D$, which represents the feature vector for a given frame image with dimensionality D (We extract deep features from the layer prior to the last fully connected layer.), $M_{deep}^{ik} \in \mathbb{R}^{1\times 1}$, and $M_{deep} \in \mathbb{R}^{n_v \times K}$. The deep distance is then defined as:

$$d_{deep} = \min_r M_{deep}. \tag{5}$$

Again, d_{deep} has the size of $n_v \times 1$. This deep distance can represent feature differences of the current video frame to its closest support KF. Both d_{raw} and d_{deep} can denote the distance between one video frame and support KFs. Next, we form a robust fusion distance by considering these two distances for KFE.

3.3.3. Fusion Distance

We combine the raw and deep distances in a new distance function to improve the performance of KF detection in our problem:

$$d = \alpha \hat{d}_{raw} + (1 - \alpha)\hat{d}_{deep}. \tag{6}$$

Since raw distance d_{raw} and deep distance d_{deep} have different magnitudes, we re-scale them by dividing them by the maximum distance within their respective matrices i.e., $\hat{d}_{raw} = d_{raw} / \max(d_{raw})$ and $\hat{d}_{deep} = d_{deep} / \max(d_{deep})$. The parameter α controls the weight between re-scale raw distance $\hat{d}_{raw}$ and re-scale deep distance $\hat{d}_{deep}$. With this new fusion distance d defined, the next step is to design a KF select function $\mathcal{S}$ that correctly retrieves KFs.

3.3.4. Key Frame Selection Mechanism

One straightforward way of extracting KFs is to return frames that have a distance score below a threshold. However, establishing an (arbitrary) threshold is prone to errors and redundancy. Different KFs could have very large distances to the support KFs, resulting in an incorrect KF. Alternatively, frames which have a distance below the threshold can belong to the same cow (redundancy). For example (Figure 5), the fusion distance when analyzing a video suggests four frames (circles) would be selected as KFs. However, each circle represents one cow, and only one frame is needed for the best view of the key cow teat frame. To address the redundancy problem, we propose to first sort d in ascending order, then iteratively take the first small distance frame as the KF, and then remove its nearby window of potentially $\pm R$ redundant frames. This process can be summarized in Algorithm 1. This key frame selection $\mathcal{S}$ allows us to uniquely obtain KFs from each cow in the video.

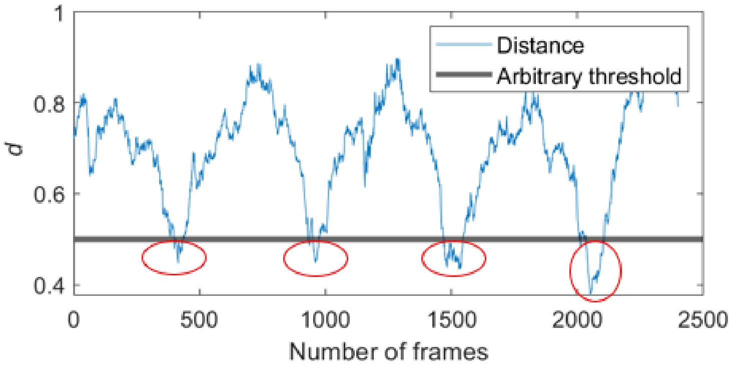


Figure 5. Threshold-based KFE. These red circle frames are the selected KFs.

Algorithm 1 Key frame selection mechanism ($\mathcal{S}$)

```
**Input:** fusion distance $d$, and redundant frame number $R = 500$
**Output:** selected key frame numbers $\mathcal{Y}_{\mathcal{S}}$
$[d_{sort}, d_{index}]$ = ascend-sort(d) // return the sorted distance and its index
$I = d_{index}$
**for** $t = 1$ **to** $len(I)$ **do**
    **if** $I_t \,! = -1$ **then**
        tem = $I_t$
        $I[(I < (I_t + R))\ \&\ (I > (I_t - R))] = -1$ // Assign $-1$ to ($\pm R$) of one key frame
        $I_t$ = tem
    **end if**
**end for**
$\mathcal{Y}_{\mathcal{S}} = unique(I)$ // Get unique key frame numbers
$\mathcal{Y}_{\mathcal{S}}[\mathcal{Y} == -1] = []$ // Remove $-1$ from the predicted KFs
return $\mathcal{Y}_{\mathcal{S}}$
```

3.3.5. Predicted KFs Quality Check

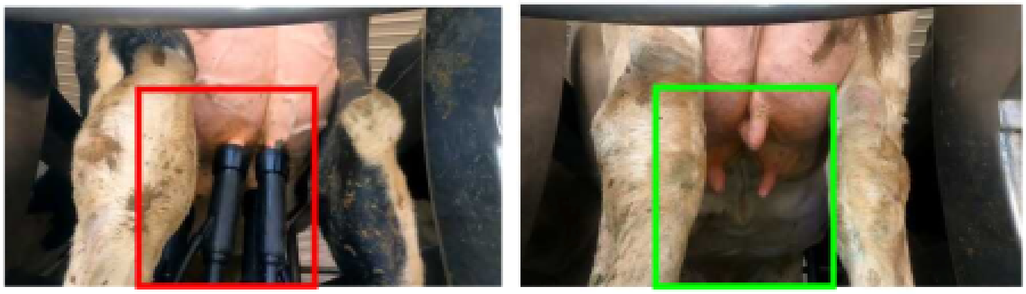
After generating several KF candidates $\mathcal{Y}_{\mathcal{S}}$ with Algorithm 1, we conduct a quality check (QC) of the predicted KFs. The most common issue for an incorrect KF candidate is when the milking unit is still attached to the dairy cow, or it obstructs visualization of the dairy cow teats (as shown in Figure 6a). To enforce selected KFs with a clear view of the teat area, we calculate the structural similarity index (SSIM) [41] score between support KF approximate teat area and selected KFs area (The position and size of teat areas is x coordinate = 130, y coordinate = 80, width = 170, height = 190, and remains constant since the camera is in a fixed position.). If the SSIM score between the most similar support KF and the selected KF is smaller than the threshold ($O = 0.45$), the selected KF is excluded. This threshold is determined empirically. Let L be the number of selected KFs candidates $\mathcal{Y}_{\mathcal{S}}$ and $\mathcal{Y}^{l}_{\mathcal{S}}$ be its l-th KF number. We then can calculate SSIM between each selected KF and each support KFs in the teat position to form a similarity matrix $H \in \mathbb{R}^{L \times K}$. An element in H is defined as

$$H_{lk} = SSIM(s^{p}_{k}, v^{p}_{\mathcal{Y}^{l}_{\mathcal{S}}}), \tag{7}$$

where p represents the sub-region of the image of greatest clinical relevance, and $v_{\mathcal{Y}^{l}_{\mathcal{S}}}$ is the selected KF image. Finally, we determine the KFs numbers with the following equation,

$$\mathcal{Y} = \mathcal{Y}_{\mathcal{S}}^{(\max\limits_{r} H) \geq O}, \tag{8}$$

where $\max\limits_{r}$ returns the maximum number of each row of the similarity matrix H. The superscript $(\max\limits_{r} H) \geq O$ selects the frame number when the highest SSIM score is greater than the threshold O.



(**a**) Detected key frame

(**b**) The closest support key frame

Figure 6. Quality check between detected key frame (**a**), which shows the milking apparatus still attached to the dairy cow, and its closest support frame (**b**). SSIM are computed over a fixed region of interest within the frame (red and green rectangles). Key frame (**a**) does not pass the quality check since its SSIM score is lower than the predetermined threshold.

Figure 6a displays a candidate KF image from $\mathcal{S}$ but the milking unit is still attached to the dairy cow teats. To mitigate this issue, we calculate the SSIM between the the current KF and support KFs within the sub-region using Equation (7), and the highest SSIM scores among all K frames to obtain its most similar support KF (Figure 6b). The SSIM score is 0.41, which is lower than the threshold $O = 0.45$. Using this method, we are able to exclude the detected KF in Figure 6a.

3.4. Ufskfe Model

Figure 4 depicts the overall framework of our proposed UFSKFE model. Combining all steps in Section 3.3, our UFSKFE model is denoted by the function:

$$\mathcal{Y} = QC(\mathcal{S}(d)), \tag{9}$$

where QC is the quality check, $\mathcal{S}$ is the section mechanism in Algorithm 1, and d is the fusion distance. The overall learning algorithm is shown in Algorithm 2.

Algorithm 2 Unsupervised few-shot key frame extraction

```
Input: Cow teat video $V$, $K = 32$ support KFs, weight balance factor $\alpha$, redundant frame number $R$ and similarity threshold $O$
Output: predicted KFs $\mathcal{Y}$
for $i = 1$ to $n_v$ do
    for $k = 1$ to $K$ do
        Compute $M_{raw}^{ik}$ and $M_{deep}^{ik}$ according to Equations (2) and (4)
    end for
end for
Calculate $d_{raw}$ and $d_{deep}$ according to Equations (3) and (5) and form $d$ using Equation (6)

Select KFs candidates using Algorithm 1
for $l = 1$ to $len(\mathcal{Y}_{\mathcal{S}})$ do
    for $k = 1$ to $K$ do
        Compute similarity matrix $H$ according to Equation (7)
    end for
end for
Return predicted key frame numbers $\mathcal{Y}_{\mathcal{S}}$ using Equations (8)
```

4. Experiments

4.1. Datasets

4.1.1. Data Collection

Approximately eight hours of video footage of dairy cow teats on a commercial dairy farm were obtained using a GoPro 10 camera mounted on a tripod with two adjustable LED lights directed towards the teats. The farm houses approximately 1600 Holstein cows which are milked daily on a 60-stall rotary parlor. The 1691 Holstein dairy cows were housed in free-stall pens and milked three times per day in a 60-stall rotary parlor. Cows were in the first (697, 41.2%), second (446, 26.4%), and third or greater lactation (548, 32.4%) and between 1 and 738 days in milk (mean and standard deviation, 185 (113)). All procedures were reviewed and approved by the Cornell University Institutional Animal Care and Use Committee (protocol no. 2013-0064). Videos were sampled at $1080\times$ 1920×3 pixels, 59.94 frames per second and saved in MP4 format. The camera was set to use default settings, and external lighting was used. The images were acquired immediately after removal of the milking cluster.

The rotational speed of the milking rotary parlor was 8.5 s/stall, leading to a rotation time of 510 s (i.e., 8.5 min). This resulted in a theoretical throughput of 423 cows per hour. The average milking duration to milk the 1600 cows was approximately five hours. The speed of rotation of the milking parlor platform does not affect the accuracy of the camera measurements, provided that the video feed is sampled at a sufficiently high enough rate. Our data were sampled with a minimum of 60 frames per second. Four milking technicians operated the milking parlor and were assigned to four different positions, including the following tasks: Position 1, manual forestripping of teats and application of pre-milking teat disinfection; position 2, cleaning and drying of teats with a clean cloth towel; position 3, attachment and alignment of the milking unit; and position 4, application of post-milking teat disinfectant with a dip-applicator cup. Post-milking teat disinfectant was applied by an automatic teat spray robot. Cows were led to the holding area by one farm technician.

Plastic covers protected the tripod and lights and were mounted around the camera to minimize the contamination of feces and other contaminants. The camera feed was displayed continuously and regularly checked to ensure that the lens was not obfuscated from such contaminants, and the camera lens itself was regularly inspected and cleaned throughout the data collection.

4.1.2. Data Analysis

Table 1 shows the statistics of the cow teat videos analyzed in this study. There are only few KFs in each cow teat video, which leads to the difficulty of KFE. Note that cows do not always occupy all the stalls in the rotating parlor, which explains why fewer key frames are detected in videos 1–10. Note also the videos are relatively large in file size (2.47 gigabytes on average), with 21,191 frames in each video. Here, the number of KFs are checked with an expert for evaluation purposes. There are usually 500 frame differences between two successive KFs unless the parlor rotation is interrupted, the parlor stall is empty, or the milking system obfuscates the teats: for these reasons, the redundant frame number R is set to 500. The computation time should be as short as reasonably possible, as it may result in delays in assessing a cow's teat health. We expect the computation time of any KFE algorithm to be less than an hour per video, which is reasonable in a commercial dairy farm setting.

Table 1. Statistics of cow teat videos (M: megabyte, G: gigabyte).

#	Video Name	# Frames	# Key Frames	Memory Size
1	GH060066	3192	6	382 M
2	GH010063	3985	7	478 M
3	GH010067	3474	3	416 M
4	GH020070	4759	7	570 M
5	GH010069	5395	7	647 M
6	GH010068	7043	10	844 M
7	GH010065	16,881	27	1.97 G
8	GH020071	24,399	30	2.85 G
9	GH030072	25,567	27	2.99 G
10	GH040066	31,860	33	3.72 G
11	GH010071	31,860	42	3.72 G
12	GH030066	31,860	44	3.72 G
13	GH010070	31,860	47	3.72 G
14	GH020072	31,860	47	3.72 G
15	GH010066	31,860	43	3.72 G
16	GH020066	31,860	48	3.72 G
17	GH010072	31,860	48	3.72 G
18	GH050066	31,860	43	3.72 G
Ave	-	21,191	29	2.47 G

4.2. Evaluation Metric

We use the F score to evaluate the performance of KFE models [27,29]. The F score uses recall (Re) and precision (Pr) to measure how much the KFs overlap in Equation (10). The higher these metrics, the better the model is.

$$Re = \frac{N_{corr}}{n_f},\ Pr = \frac{N_{corr}}{len(\mathcal{Y})},\ F = \frac{2Re \times Pr}{Re + Pr}, \tag{10}$$

where n_f is the number of ground truth KFs (third column in Table 1), $len(\mathcal{Y})$ is the length of the predicted KFs, and N_{corr} is the number of correctly detected KFs. The closer the F score is to 1 (or 100% in Table 2), the better the model. Since several nearby frames of annotated KFs are similar to each other, they also contain a clear view of the teat area. Therefore, we treat the annotated KF number within ±20 frames (or approximately 0.3 s) as the correct prediction (e.g., a predicted KF number of 120 is correct if the annotated KF number is 100). This value will vary based on the video frame rate and rotation rate of the parlor.

Table 2. F score (%) and computation time (s) of cow teat video key frame extraction (QC is conducted, NAST: NASNetLarge).

Videos	d_{SURF}		d_{Binary}		d_{Sobel}		H_{SSIM}		d^{crop}_{raw}		d_{raw}		$d^{AlexNet}_{deep}$		d^{NAST}_{deep}		$d^{ResNet-101}_{deep}$		**UFSKFE**	
	F	**Time**	**F**	**Time**	**F**	**Time**	**F**	**Time**	**F**	**Time**	**F**	**Time**	**F**	**Time**	**F**	**Time**	**F**	**Time**	**F**	**Time**
GH060066	54.5	109.5	72.7	66.9	40.0	27.6	72.7	3435.2	20.0	84.9	72.7	41.1	72.7	**14.4**	60.0	57.7	**90.9**	20.8	**90.9**	65.3
GH010063	57.1	118.6	57.1	81.0	42.9	40.2	14.3	1038.1	0.0	105.8	57.1	51.0	30.8	**19.0**	42.9	94.0	28.6	20.4	**61.5**	84.5
GH010067	22.2	102.6	44.4	71.8	85.7	42.3	**66.7**	902.5	0.0	92.2	44.4	44.2	22.2	18.0	44.4	73.1	60.0	**17.7**	50.0	71.3
GH020070	26.7	142.4	**62.5**	100.1	30.8	57.8	25.0	1241.6	37.5	127.5	50.0	61.1	12.5	**21.6**	0.0	99.9	37.5	23.0	61.5	97.5
GH010069	23.5	160.7	**70.6**	110.1	53.3	65.9	23.5	1408.2	23.5	148.6	58.8	69.3	47.1	**23.9**	0.0	122.4	58.8	26.5	66.7	110.4
GH010068	34.8	228.8	45.5	144.0	20.0	84.2	43.5	1830.9	9.1	188.4	69.6	90.1	52.2	**32.3**	34.8	158.2	69.6	34.2	**80.0**	145.8
GH010065	28.6	639.1	45.6	346.1	46.2	219.8	10.7	4395.2	15.4	453.2	40.7	218.9	17.5	**84.2**	35.1	354.4	42.1	86.7	**50.0**	359.7
GH020071	34.4	1161.7	58.3	507.3	39.4	364.7	43.2	6442.8	2.9	651.8	57.5	348.5	19.2	**145.3**	33.3	554.0	43.8	188.0	**65.6**	544.2
GH030072	30.8	1183.4	43.8	562.7	34.0	219.1	38.9	6756.0	3.1	653.4	39.4	375.4	25.0	**181.3**	16.7	593.7	44.8	181.5	**55.6**	581.9
GH040066	69.7	1151.0	68.9	695.6	70.5	308.8	67.4	8395.9	30.6	590.2	71.1	478.9	69.7	280.3	71.9	810.6	74.2	**268.7**	**75.9**	780.6
GH010071	34.0	1170.4	58.6	697.3	42.7	303.9	34.3	8710.8	19.6	584.5	56.8	475.5	30.9	299.4	30.6	807.6	53.1	**272.9**	**61.9**	792.4
GH030066	18.8	1145.3	51.5	671.8	38.8	312.2	48.0	8794.7	23.9	579.5	**53.5**	462.8	43.1	**282.0**	37.6	777.9	49.0	289.0	52.1	784.3
GH010070	35.6	1169.2	**55.2**	682.8	29.5	305.0	36.2	8733.1	18.6	581.4	50.5	476.5	23.3	311.8	45.7	799.3	34.3	**277.9**	54.5	783.0
GH020072	27.7	1176.1	53.3	666.0	37.8	338.7	25.5	9812.4	23.2	588.6	**53.8**	475.9	23.1	408.7	29.7	818.0	46.2	**281.8**	51.6	767.8
GH010066	19.1	631.6	56.9	667.1	44.2	345.5	43.3	9709.1	30.9	598.1	52.5	486.0	36.0	327.3	43.6	799.4	48.5	**281.9**	**70.2**	808.5
GH020066	21.4	671.6	50.5	667.4	53.3	333.7	39.3	11023.7	38.0	589.5	**58.5**	487.3	34.3	300.4	35.5	804.0	**58.5**	**285.2**	55.8	792.3
GH010072	18.4	710.2	50.9	673.6	40.9	329.4	26.7	10083.3	22.0	587.9	52.8	474.0	17.1	293.5	36.2	780.3	37.4	**286.7**	**66.0**	786.0
GH050066	24.7	661.0	47.5	681.8	44.4	327.7	58.6	8243.1	34.4	588.1	57.1	478.5	47.5	**277.6**	25.7	785.5	59.6	286.1	**74.2**	804.3
Ave	32.3	685.2	55.3	449.6	44.1	223.7	39.9	6164.3	19.6	433.0	55.4	310.8	34.7	184.5	34.7	516.1	52.1	**173.8**	**63.6**	508.9

4.3. Implementation Details

In our UFSKFE model, we utilize ResNet-101 [42] as the pre-trained model to extract deep features from the layer prior to the last fully connected layer. We conducted experiments with 12 different ImageNet models in order to justify selecting ResNet-101 for feature extraction. The performance of different ImageNet models can be found in Appendix A. Frame image features are extracted with an NVIDIA RTX A6000 GPU with 48 Gigabyte. The three hyperparameters are set at $\alpha = 0.4$, $R = 500$ and $O = 0.45$. We also conduct a parameter analysis in Section 4.6. Since there are no KFE models to adopt in our problem, we compare several existing models with different frame image extraction methods. In Section 3.3.2, Φ refers to the feature extractor from an ImageNet model. We can also extract other features, such as SURF features [9,10], binary image features [43] and Sobel edge detection image features [44]. We then can calculate d_{SURF}, d_{Binary} and d_{Sobel}. We replace the fusion distance d with these other distances in Algorithm 1 to predict KFs. The details of feature extraction can be found in Appendix A.1. Results in Table 2 are reported with the additional quality check.

4.4. Results

Table 2 shows the performance of all 18 cow teat videos. Compared with all other baselines, our UFSKFE model achieves the highest average F score over all videos. Note that while d^{crop}_{raw} has the lowest F score, it only calculates the distance between each video frame to the support KF images in the small teat area, and ignores other important areas, e.g., the cow leg area. We find that the performance of extracted AlexNet [45] features and NASNetLarge [46] features are similar and lower than ResNet-101 [42] features. One reason for this is that AlexNet is not a high-performance ImageNet model. Its extracted features might focus on shallow features. In comparison, the performance of the NASNet-Large model is high, suggesting extracted features might lead to ImageNet image features. The F score of the SURF features is lower than those of the AlexNet and NASNetLarge features, likely because the SURF features can only detect a few important points while ignoring background features. d_{raw} achieves the second-best results, which demonstrates that the raw images also contain important features that deep neural networks are not captured. The performance when using binarized images of the videos is similar to the performance of d_{raw} since it contains similar important features to the raw frame images. They both perform better than the Sobel edge detection images likely because edge features are predominantly analyzed. In terms of computation time, deep feature extraction with ResNet-101 is faster than with all other models. Although our UFSKFE model takes longer, it combines the feature extraction time of ResNet-101 and raw images. The total average time to extract KFs is less than nine minutes and is faster than extracting SURF, NASNetLarge features, and H_{SSIM} (details are shown in Appendix A.2). The SSIM similarity selection is not an efficient method, with computation times of more than 1.7 h per video. These extensive results demonstrate that our proposed UFSKFE model can quickly and accurately extract KFs.

Figure 7 shows KFs detection using our model from the GH060066 video using fusion distance d. There are five true KFs (green dots), while our model detected six points as the KFs (red dots). Although there are differences between green and red dots, those differences are within ±20 frames, and are considered correct predictions. Figure 8 compares detected KF images with the ground truth. In our UFSKFE model, only one wrong prediction (frame 2611) is detected. This is likely due to the milking apparatus still being attached to the dairy cow's teats with the low field of view. The quality check process does not remove this detected KF (similarity score 0.62 exceeds threshold O). The other two methods d_{SURF} and d_{Binary} also incorrectly identify this cow's images as a key frame. Compared with predicted KF images of other models, UFSKFE has a higher F score, and these frames are closer to the ground truth KF images than other models.

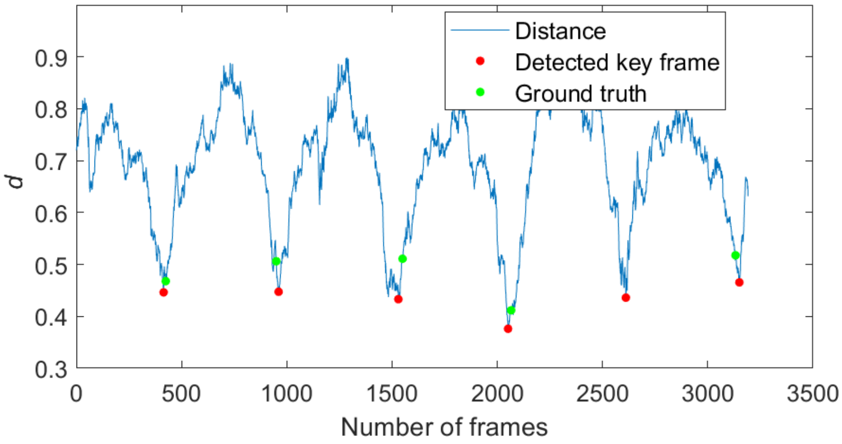


Figure 7. Extracted key frame numbers with our UFSKFE model of GH060066 cow teat video. d is the fusion distance to the supported KFs. Red dots are the detected KFs, while green dots are true KFs.

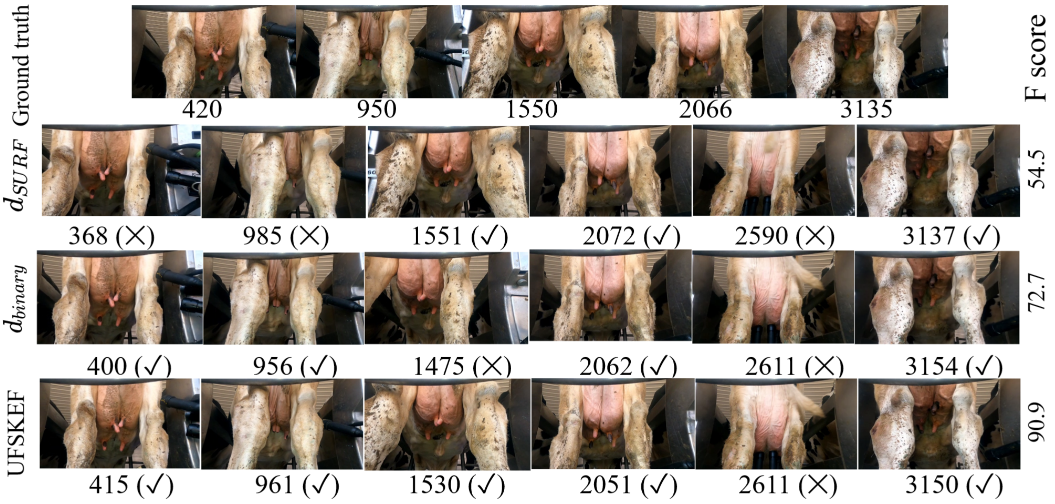


Figure 8. Different key frame comparisons of GH060066 video. The ✓means a correct prediction, while × means a wrong prediction. The number below each image is the video frame. The F score is also reported in each method. UFSKFE achieves the highest F score.

4.5. Ablation Study

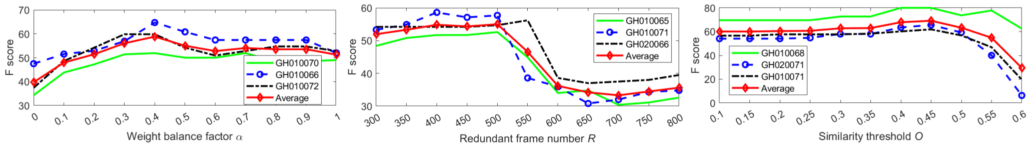
To demonstrate the effectiveness of different components on the final F score, we conduct an ablation study for each component of our proposed UFSKFE model (Table 3) with four randomly selected videos (GH060066, GH030072, GH010066, and GH050066). Realizing that the KF selection function $\mathcal{S}$ is required, we conduct the ablation study with d_{raw}, d_{deep}, and QC. d_{raw} selects the KFs using raw distance with $\mathcal{S}$, d_{deep} selects the KFs using deep distance with $\mathcal{S}$, and $d_{raw} + QC$ conducts a quality check after selecting the KFs of the raw distance. We find that the F score for fusion distance d is higher than when using d_{raw} or d_{deep}. The quality check process is also effective in improving the F score. Therefore, all our proposed components demonstrate effectiveness and importance in this KFE.

Table 3. F score (%) of ablation study.

Videos	GH060066	GH030072	GH010066	GH050066	Ave
d_{raw}	72.7	38.4	52.0	56.0	54.8
d_{deep}	72.7	40.5	47.5	54.9	53.9
$d_{raw} + QC$	72.7	39.4	52.5	57.1	55.4
$d_{deep} + QC$	**90.9**	44.8	48.5	59.6	61.0
d	**90.9**	45.3	64.7	64.7	66.4
UFSKFE	**90.9**	**55.6**	**70.2**	**74.2**	**72.7**

4.6. Parameter Analysis

There are three hyperparameters in our model: weight balance factor α, redundant frame number R and similarity threshold O. To determine the best parameters, we report F score of randomly select three videos when these hyperparameters are varied. α is selected from $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.1\}$, $R \in \{300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800\}$ and $O \in \{0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6\}$. We vary each parameter independently, while keep others fixed. From Figure 9a,c, we find that when $\alpha = 0.4$, $R = 500$ and $O = 0.45$, the F score is maximized.



(**a**) Effects of α on F score (**b**) Effects of R on F score (**c**) Effects of O F score

Figure 9. Parameter analysis for α, R, and O on F score. When $\alpha = 0.4$, $R = 500$ and $O = 0.45$, the average F score achieves the maximum number.

5. Discussion of Results and Limitations of UFSKFE Model

Our UFSKFE achieves the highest average F score when compared with other methods. There are three possible reasons why this model performs well. First, the proposed unsupervised few-shot learning paradigm leverages knowledge from a few support KFs to all of the video frames. Second, our proposed fusion distance takes advantage of both raw and deep distances from support frames that represent a diverse range of possible key frames. Third, the quality check process acts as an effective method for removing noisy KF candidates, resulting in a substantially improved overall performance.

A limitation of our proposed UFSKFE model is that it cannot remove some of the KFs, primarily those images where the milking apparatus remains attached to the dairy cows. Although our quality check can remove some of these images (Figure 6a) the removal of such images becomes more challenging when the cameras' field of view does not adequately image the cows' teats. This can be circumvented with re-positioning the camera in the portrait as opposed to landscape mode when collecting the videos. In addition, we only extract one key frame per cow. A clear view of each teat in the same cow can come from different frames. Therefore, we can consider extracting key frames for each teat. Exploring other methods for extracting features from video frames from few-shot learning may also be of value in our efforts to improve performance. Furthermore, our process can be performed in real time if we can directly store the recorded video on the cloud console.

Future work focuses on the use of other machine learning approaches to assess the extent of hyperkeratosis and the risk of mastitis.

The performance of our key-frame extraction methodology may also be influenced by farm- and cow-related factors. With regards to the farm itself, the lighting conditions and cleanliness of the farm, stall, and parlors could affect the performance. The rotary parlor is housed inside a large complex which mitigates the effects of weather, lighting, and other environmental factors that could affect the quality of the video data. Variations from best milking practices could similarly affect performance, such as inconsistent cleaning of the teat ends. Key-frames with the milking unit still attached to the cow will depend on the settings of the milking system (vacuum pressure and detachment), parlor rotational speed, and location of the camera. Finally, the performance of the key-frame extraction method will depend on the size of the dataset. Our model was developed using only 32 labeled key frames. While additional labeled data could improve overall performance, our findings suggest that in the commercial dairy farm setting where such rotating parlors are used, key frames from only a very small fraction of the herd are necessary if using our automated key-frame extraction technique.

6. Conclusions

In this paper, we propose a novel unsupervised few-shot learning key frame extraction model for cow teat videos. We combine the raw and deep distances between each video frame and support key frame images and form a fusion distance to better denote the differences between each video frame and support key frame images. An efficient key frame selection mechanism is proposed to first determine the key frame candidates, followed by a quality check procedure to refine the predicted key frames. Extensive experiment results demonstrate that the proposed UFSKFE model can accurately and efficiently extract the key cow teat frames. Our approach provides an opportunity to reduce the redundancy of processing large videos. The extracted key teat-end frames can be collected to monitor the health status of dairy cows.

Author Contributions: Conceptualization, Y.Z. and P.S.B.; methodology, Y.Z.; writing—original draft preparation, Y.Z.; writing—review and editing, P.S.B. and M.W.; supervision, P.S.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Cornell Initiative for Digital Agriculture (CIDA).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Source code and datasets will be at https://github.com/YoushanZhang/UFSKFE (accessed on 17 May 2022).

Acknowledgments: The authors thank the farm owners and their employees for their willingness to participate in the study and their support during the data collection.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Appendix A.1

To calculate deep distance, we need to extract deep features for video frames from pre-trained ImageNet models. To select the best ImageNet model for feature extraction of our cow teat videos, we conducted extensive experiments with 12 frequently used ImageNet models. We use the layer prior to the last fully connected layer to extract deep features. These 12 ImageNet models are AlexNet [45], VGG16 [47], VGG19 [47], GoogLeNet [48], DenseNet-201 [49], ResNet-18 [42], ResNet-50 [42] , ResNet-101 [42] , Inception-V3 [50], Xception [51], InceptionResNet-V2 [52], NASNetLarge [46].

We also utilize t-SNE [53] to visualize extracted deep features in 2D space as shown in Figure A1, but it is still difficult to select the best pre-trained ImageNet model. We thus plot the projected loss from a high dimension space to the 2D space of the t-SNE model in Figure A2. We observe that ResNet-101 has the smallest projection loss among the 12 models, which suggests that ResNet-101 is a suitable ImageNet model for extracting deep features. However, we still do not know whether the performance of ResNet-101 features of our key frame extraction problem is better than other deep features. We thus report the performance of all 12 models in Table 2. We first calculate these deep distances and use key frame selection $\mathcal{S}$ to select the key frame candidates, then the quality check is performed to remove noisy key frames. We can find that the deep ReseNet-101 distance indeed achieves a higher F score than other models.

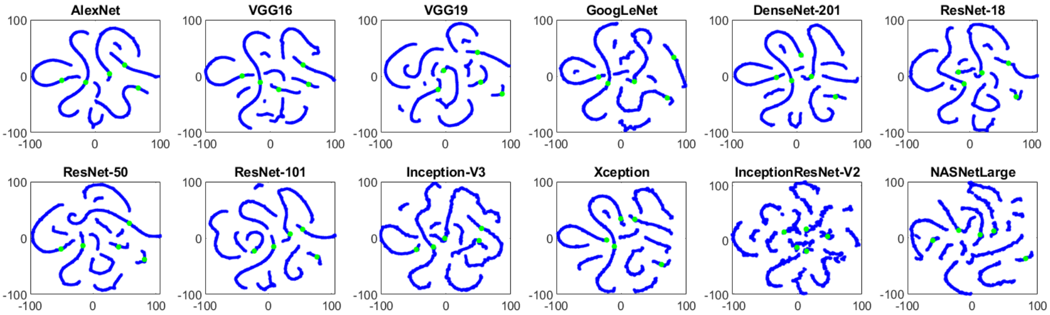


Figure A1. T-SNE visualization of extracted features of 12 ImageNet models from GH060066 video. Blue represents video frames, while green dots are the key frame image position.

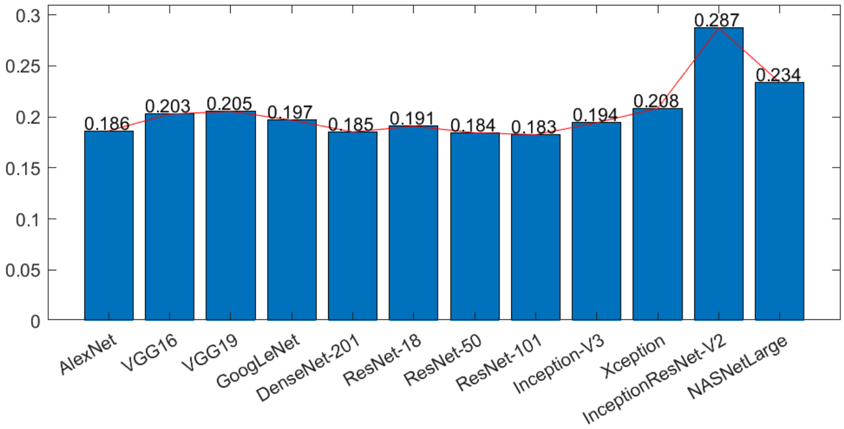


Figure A2. T-SNE projection loss of different ImageNet models. Y-axis denotes the projected loss from high dimension space to the 2D space.

Appendix A.2. Other Baseline Features

Here, we provide the details of extracting features from other baselines. As shown in Figure A3, we show SURF detected points, binary image, and edge detection using the Sobel algorithm. In Figure A3b, we only show 10 of the strongest SURF points, while a total of 500 points are extracted from each video frame and supported key frame images, and there are 64 features for each point. We could extract $500 \times 64 = 3200$ SURF features for each image. The SURF distance is defined as follows:

$$d_{SURF} = \min_{r} M_{SURF}, \; M^{ik}_{SURF} = |\phi(s_k) - \phi(v_i)|_1, \tag{A1}$$

where ϕ refers to the SURF feature extractor and $\phi(\cdot) \in \mathbb{R}^{1\times 3,200}$. In Figure A3c, we calculate the distance between the binary image of each video frame and support key frame images. The binary distance is defined as follows:

$$d_{Binary} = \min_{r} M_{binary}, \; M^{ik}_{binary} = |B(s_k) - B(v_i)|_1, \tag{A2}$$

where B refers to obtaining the binary image. In Figure A3d, we calculate the distance between edge detection images using the Sobel algorithm of each video frame and support key frame images. The Sobel distance is defined as follows:

$$d_{Sobel} = \min_{r} M_{Sobel}, \ M^{ik}_{Sobel} = |E(s_k) - E(v_i)|_1, \tag{A3}$$

where E refers to obtaining an edge detection image using the Sobel algorithm. After obtaining SURF distance d_{SURF}, binary distance d_{Binary} and Sobel distance d_{Sobel}, we use the key frame selection $\mathcal{S}$ (the fusion distance d will be replaced with these three distances, respectively) and perform the quality check process to obtain the final extracted key frames.

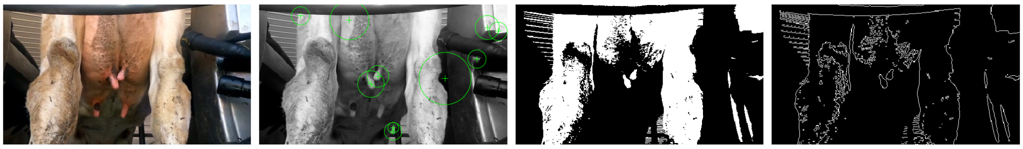

(**a**) Raw image (**b**) SURF points (**c**) Binary image (**d**) Sobel edge

Figure A3. Raw image, SURF 10 strongest points, binary image edge detection with Sobel algorithm image comparison.

We utilize visualize extracted SURF features (Figure A4) using t-SNE. These frame features (blue dots) are indistinguishable from the non-key frames, as shown in blue in Figure A1. SURF features also have a higher project loss (1.819) than other ImageNet models. This implies that the performance of SURF features might be lower than different ImageNet models. The average F score of SURF is 32.3 (in Table 1 of the main paper), which is lower than most ImageNet models.

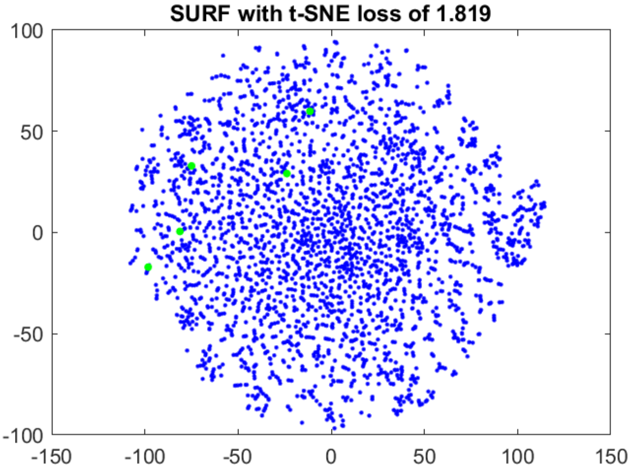


Figure A4. T-SNE visualization of extracted SURF features.

In Table 1 of the main paper, we also present the result of H_{SSIM}, using the SSIM similarity matrix to determine key frames. Specifically, we calculate the SSIM score of the crop teat area of each video frame image and support video key frame image, and then select the highest score to detect the key frames and then perform the quality check process. The SSIM similarity matrix is defined as follows:

$$H_{SSIM} = \max_{r} h_{SSIM}, \ h^{ik}_{SSIM} = SSIM(s^p_k, v^p_i), \tag{A4}$$

where p represents the teat position area, and $\max_{r}$ returns the maximum number of each row of the similarity matrix $h_{SSIM} \in \mathbb{R}^{n_v \times K}$. Hence, $H_{SSIM} \in \mathbb{R}^{n_v \times 1}$. We then have a partial new key frame selection function $\mathcal{S}'$ to determine the key frame candidates as in Algorithm A1. There are two changes, the first is that the input is not the fusion distance d, but the SSIM similarity matrix H_{SSIM}. Secondly, we sort H_{SSIM} in a descending order since a more similar teat area is more likely to be a key frame. Figure A5 shows the process of detecting key frames using new key frame selection function ($\mathcal{S}'$) on GH060066 video with similarity matrix H_{SSIM}.

Algorithm A1 Key frame selection mechanism ($\mathcal{S}'$)

```
Input: SSIM similarity matrix $H_{SSIM}$, and redundant frame number $R = 500$
Output: selected key frame numbers $\mathcal{Y}_{\mathcal{S}}$
$[H_{sort}, H_{index}]$ = descend-sort($H_{SSIM}$) // return the sorted distance and its index
$I = H_{index}$
for $t = 1$ to $len(I)$ do
    if $I_t\ != -1$ then
        tem = $I_t$
        $I[(I < (I_t + R))\ \&\ (I > (I_t - R))] = -1$ // Assign $-1$ to ($\pm R$) of one key frame
        $I_t$ = tem
    end if
end for
$\mathcal{Y}_{\mathcal{S}} = unique(I)$
$\mathcal{Y}_{\mathcal{S}}[\mathcal{Y} == -1] = []$ // Remove $-1$ from the predicted key frames
return $\mathcal{Y}_{\mathcal{S}}$
```

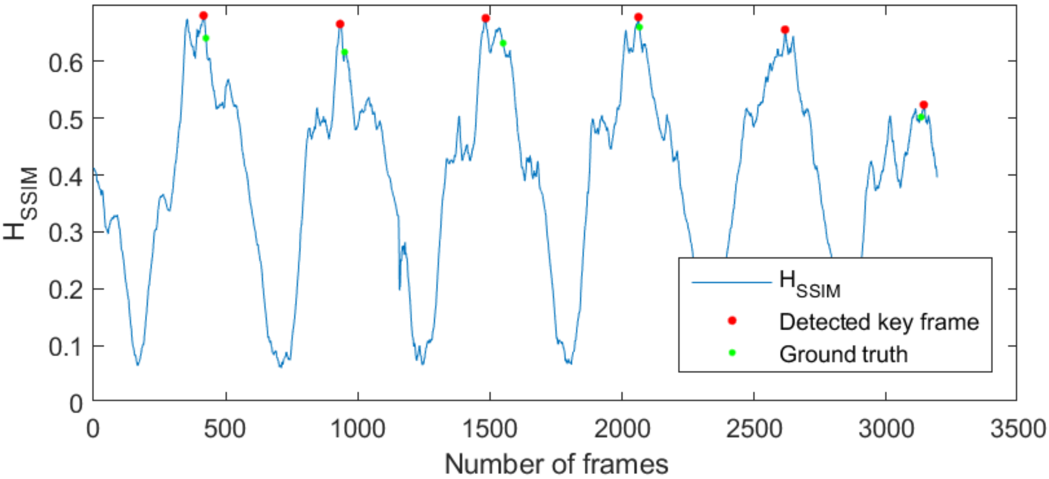


Figure A5. Key frame extraction with H_{SSIM} model of GH060066 cow teat video. H_{SSIM} is similarity matrix. Red dots are the detected key frames, while green dots are the ground truth key frames.

Appendix A.3. Other Ablation Study

In this section, we show small variants of our UFSKFE model. As shown in Table A2, $d_{deep}^{ResNet-101-p}$ refers to calculating the deep distance using the crop teat area position. $L2$ norm means balancing the scale of raw distance and deep distance using L2 norm. In the main paper, we use $\hat{d}_{raw} = d_{raw} /\max(d_{raw})$ and $\hat{d}_{deep} = d_{deep} /\max(d_{deep})$ to balance the scale between them. Here, we instead use the L2 norm, i.e., $\hat{d}_{raw} = d_{raw}/||d_{raw}||_2$ and $\hat{d}_{deep} = d_{deep}/||d_{deep}||_2$. The raw $L2$ distance means that we calculate the L2 distance in Equation (2) of the main paper. It can be denoted as $M_{raw}^{ik} = ||s_k - v_i||_2$.

Table A1. F score (%) and computation time (s) of 12 different ImageNet models (IR: InceptionResNet-V2, NAST: NASNetLarge).

Videos	$d_{deep}^{AlexNet}$		d_{deep}^{VGG16}		d_{deep}^{VGG19}		$d_{deep}^{GoogLeNet}$		$d_{deep}^{DenseNet-201}$		$d_{deep}^{ResNet-18}$	
	F	**Time**	**F**	**Time**	**F**	**Time**	**F**	**Time**	**F**	**Time**	**F**	**Time**
GH060066	72.7	14.4	72.7	25.9	36.4	28.4	18.2	11.8	36.4	42.8	54.5	21.3
GH010063	30.8	19.0	14.3	30.5	28.6	31.0	28.6	15.7	15.4	52.4	14.3	15.6
GH010067	22.2	18.0	60.0	27.0	20.0	27.6	66.7	14.1	22.2	46.3	60.0	14.2
GH020070	12.5	21.6	37.5	35.6	37.5	36.5	12.5	17.8	37.5	62.1	37.5	18.0
GH010069	47.1	23.9	58.8	40.4	47.1	40.7	58.8	19.9	47.1	70.6	35.3	19.8
GH010068	52.2	32.3	27.3	56.9	43.5	61.8	43.5	28.6	52.2	101.9	43.5	25.0
GH010065	17.5	84.2	38.6	137.8	45.6	134.3	25.0	67.3	17.5	233.5	38.6	65.3
GH020071	19.2	145.3	33.3	228.1	40.5	229.4	31.0	132.3	33.3	424.3	38.4	152.3
GH030072	25.0	181.3	8.2	228.0	32.9	242.0	22.9	142.0	25.0	428.4	35.1	157.3
GH040066	69.7	280.3	74.2	333.6	71.1	359.4	70.5	255.0	71.9	526.4	75.0	248.7
GH010071	30.9	299.4	34.7	338.3	47.4	341.9	29.2	253.2	20.6	509.7	42.4	249.2
GH030066	43.1	282.0	33.7	322.4	39.6	343.0	33.7	242.1	26.0	505.8	51.5	249.2
GH010070	23.3	311.8	36.2	331.1	45.7	348.8	19.6	229.3	40.4	516.8	45.7	248.6
GH020072	23.1	408.7	27.2	365.4	32.4	334.9	13.6	243.3	37.3	518.6	42.3	252.1
GH010066	36.0	327.3	41.6	325.4	45.1	338.0	33.3	247.8	31.7	495.6	45.1	252.0
GH020066	34.3	300.4	44.9	326.1	45.3	329.8	41.9	292.2	39.3	510.0	41.5	252.1
GH010072	17.1	293.5	31.8	327.3	44.9	335.6	15.1	293.2	27.2	511.3	44.9	241.7
GH050066	47.5	277.6	54.9	330.8	54.9	347.1	21.8	264.9	25.7	511.9	47.1	250.2
Ave	34.7	184.5	40.6	211.7	42.1	217.2	32.6	153.9	33.7	337.1	44.0	151.8

Videos	$d_{deep}^{AlexNet}$		d_{deep}^{VGG16}		d_{deep}^{VGG19}		$d_{deep}^{GoogLeNet}$		$d_{deep}^{DenseNet-201}$		$d_{deep}^{ResNet-18}$	
	F	**Time**	**F**	**Time**	**F**	**Time**	**F**	**Time**	**F**	**Time**	**F**	**Time**
GH060066	54.5	14.3	90.9	20.8	72.7	22.5	18.2	29.5	36.4	30.9	60.0	57.7
GH010063	14.3	18.0	28.6	20.4	14.3	28.0	14.3	37.7	0.0	36.3	42.9	94.0
GH010067	60.0	16.0	60.0	17.7	60.0	24.7	60.0	33.2	22.2	33.2	44.4	73.1
GH020070	50.0	20.9	37.5	23.0	40.0	33.2	13.3	44.4	40.0	43.9	0.0	99.9
GH010069	47.1	22.8	58.8	26.5	12.5	39.5	35.3	50.2	37.5	49.2	0.0	122.4
GH010068	52.2	40.3	69.6	34.2	60.9	53.7	17.4	66.4	17.4	65.3	34.8	158.2
GH010065	45.6	80.9	42.1	86.7	38.6	121.2	28.1	161.0	18.2	159.5	35.1	354.4
GH020071	32.9	140.4	43.8	188.0	28.2	218.0	28.6	250.4	14.1	268.5	33.3	554.0
GH030072	32.0	155.9	44.8	181.5	22.5	239.2	16.7	287.8	16.9	278.2	16.7	593.7
GH040066	75.0	266.7	74.2	268.7	68.9	343.4	68.2	424.3	71.9	418.9	71.9	810.6
GH010071	50.5	269.1	53.1	272.9	35.4	369.2	16.5	440.5	20.8	402.8	30.6	807.6
GH030066	39.6	300.3	49.0	289.0	30.6	342.7	31.7	435.8	28.3	423.5	37.6	777.9
GH010070	39.6	275.0	34.3	277.9	32.4	338.3	40.0	434.7	21.2	427.0	45.7	799.3
GH020072	45.7	287.1	46.2	281.8	15.8	326.8	31.1	420.8	14.0	425.6	29.7	818.0
GH010066	52.9	259.9	48.5	281.9	32.3	361.1	38.4	439.0	14.1	399.1	43.6	799.4
GH020066	50.5	248.3	58.5	285.2	39.6	360.4	39.3	421.7	19.4	422.8	35.5	804.0
GH010072	44.9	272.7	37.4	286.7	36.9	344.3	21.2	423.2	29.1	448.8	36.2	780.3
GH050066	47.1	269.2	59.6	286.1	33.3	332.2	36.4	423.0	22.4	437.2	25.7	785.5
Ave	46.4	164.3	52.1	173.8	37.5	216.6	30.8	268.0	24.7	265.0	34.7	516.1

From Table A2, we find that the performance of $d_{deep}^{ResNet-101-p}$ (31.4) is much lower than that of $d_{deep}^{ResNet-101}$ (52.1). The reason is that the small teat area tends to ignore other important background features. The performance of $L2$ norm (58.15) is also lower than the simple $L1$ norm (63.6 in Table 2). In addition, the F score of raw $L2$ distance (55.2) is slightly lower than the performance of the $L1$ distance 55.4 (Table 2). We can conclude that all proposed strategy in our UFSKFE model is effective in improving the accuracy of key frame extraction in cow teat videos.

Table A2. Ablation study of different variants of UFSKFE.

Video Name	$d_{deep}^{ResNet-101-p}$	Feature $L2$ Norm	Raw $L2$ Distance
GH060066	54.5	90.9	54.5
GH010063	14.3	57.1	57.1
GH010067	60.0	44.4	44.4
GH020070	37.5	50.0	75.0
GH010069	11.8	58.8	58.8
GH010068	26.1	69.6	69.6
GH010065	39.3	52.6	37.0
GH020071	30.6	54.1	60.3
GH030072	19.7	46.6	36.6
GH040066	73.3	73.3	68.9
GH010071	24.5	57.7	52.1
GH030066	38.0	49.0	56.9
GH010070	17.6	52.9	48.5
GH020072	25.0	48.5	53.8
GH010066	21.6	62.0	56.0
GH020066	24.5	56.6	56.6
GH010072	11.5	57.9	50.9
GH050066	35.6	64.7	57.1
Ave	31.4	58.2	55.2

Appendix A.4. Cow Teat Process Video

We attached a UFSKFE_GH060066.mp4 demo video, which demonstrates the process of our UFSKFE detecting key frames in the GH060066 video. We first plot the fusion distance and then show the extracted six key frames. The video frame is accelerated, skipping every 10 frames in the demo, which leads to the oscillation of the demo video. The fusion distance of all frames is also plotted. The actual computation time of our model for GH060066 video is 65.3 s.

References

1. Reinemann, D.; Rasmussen, M.; LeMire, S.; Neijenhuis, F.; Mein, G.; Hillerton, J.; Morgan, W.; Timms, L.; Cook, N.; Farnsworth, R.; et al. Evaluation of bovine teat condition in commercial dairy herds: 3. Getting the numbers right. In Proceedings of the 2nd International Symposium on Mastitis and Milk Quality, NMC/AABP, Vancouver, BC, Canada, 12–14 September 2001; pp. 357–361.
2. Basran, P.S.; Wieland, M.; Porter, I.R. A digital technique and platform for assessing dairy cow teat-end condition. *J. Dairy Sci.* **2020**, *103*, 10703–10708. [CrossRef] [PubMed]
3. Porter, I.R.; Wieland, M.; Basran, P.S. Feasibility of the use of deep learning classification of teat-end condition in Holstein cattle. *J. Dairy Sci.* **2021**, *104*, 4529–4536. [CrossRef] [PubMed]
4. Zhang, Y.; Porter, I.R.; Wieland, M.; Basran, P.S. Separable Confident Transductive Learning for Dairy Cows Teat-End Condition Classification. *Animals* **2022**, *12*, 886. [CrossRef]
5. Wolf, W. Key frame selection by motion analysis. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*; IEEE: Atlanta, GA, USA, 1996; Volume 2, pp. 1228–1231.
6. Kulhare, S.; Sah, S.; Pillai, S.; Ptucha, R. Key frame extraction for salient activity recognition. In *2016 23rd International Conference on Pattern Recognition (ICPR)*; IEEE: Cancun, EM, USA, 2016; pp. 835–840.
7. Guan, G.; Wang, Z.; Lu, S.; Da Deng, J.; Feng, D.D. Keypoint-based keyframe selection. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *23*, 729–734. [CrossRef]
8. Hannane, R.; Elboushaki, A.; Afdel, K.; Naghabhushan, P.; Javed, M. An efficient method for video shot boundary detection and keyframe extraction using SIFT-point distribution histogram. *Int. J. Multimed. Inf. Retr.* **2016**, *5*, 89–104. [CrossRef]
9. Luo, Y.; Zhou, H.; Tan, Q.; Chen, X.; Yun, M. Key frame extraction of surveillance video based on moving object detection and image similarity. *Pattern Recognit. Image Anal.* **2018**, *28*, 225–231. [CrossRef]
10. Yu, L.; Cao, J.; Chen, M.; Cui, X. Key frame extraction scheme based on sliding window and features. *Peer- Netw. Appl.* **2018**, *11*, 1141–1152. [CrossRef]
11. Zhuang, Y.; Rui, Y.; Huang, T.S.; Mehrotra, S. Adaptive key frame extraction using unsupervised clustering. In Proceedings of the 1998 International Conference on Image Processing, Chicago, IL, USA, 7 October 1998; Volume 1, pp. 866–870.
12. Mendi, E.; Bayrak, C. Shot boundary detection and key-frame extraction from neurosurgical video sequences. *Imaging Sci. J.* **2012**, *60*, 90–96. [CrossRef]

13. Priya, G.L.; Domnic, S. Shot based keyframe extraction for ecological video indexing and retrieval. *Ecol. Inform.* **2014**, *23*, 107–117. [CrossRef]
14. Vázquez-Martín, R.; Bandera, A. Spatio-temporal feature-based keyframe detection from video shots using spectral clustering. *Pattern Recognit. Lett.* **2013**, *34*, 770–779. [CrossRef]
15. Ioannidis, A.; Chasanis, V.; Likas, A. Weighted multi-view key-frame extraction. *Pattern Recognit. Lett.* **2016**, *72*, 52–61. [CrossRef]
16. Lee, Y.J.; Ghosh, J.; Grauman, K. Discovering important people and objects for egocentric video summarization. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 1346–1353.
17. Gygli, M.; Grabner, H.; Riemenschneider, H.; Van Gool, L. Creating summaries from user videos. In *European Conference on Computer Vision*; Springer: Zurich, Switzerland, 2014; pp. 505–520.
18. Yao, P. Key Frame Extraction Method of Music and Dance Video Based on Multicore Learning Feature Fusion. *Sci. Program.* **2022**, *2022*, 9735392. [CrossRef]
19. Zhang, K.; Chao, W.L.; Sha, F.; Grauman, K. Video summarization with long short-term memory. In *European Conference on Computer Vision*; Springer: Amsterdam, The Netherlands, 2016; pp. 766–782.
20. Zhao, B.; Li, X.; Lu, X. Hierarchical recurrent neural network for video summarization. In Proceedings of the 25th ACM International Conference on Multimedia, Mountain View, CA, USA, 23–27 October 2017; pp. 863–871.
21. Zhao, B.; Li, X.; Lu, X. TTH-RNN: Tensor-train hierarchical recurrent neural network for video summarization. *IEEE Trans. Ind. Electron.* **2020**, *68*, 3629–3637. [CrossRef]
22. Fajtl, J.; Sokeh, H.S.; Argyriou, V.; Monekosso, D.; Remagnino, P. Summarizing videos with attention. In *Asian Conference on Computer Vision*; Springer: Perth, Australia, 2018; pp. 39–54.
23. Li, P.; Ye, Q.; Zhang, L.; Yuan, L.; Xu, X.; Shao, L. Exploring global diverse attention via pairwise temporal relation for video summarization. *Pattern Recognit.* **2021**, *111*, 107677. [CrossRef]
24. Jian, M.; Zhang, S.; Wu, L.; Zhang, S.; Wang, X.; He, Y. Deep key frame extraction for sport training. *Neurocomputing* **2019**, *328*, 147–156. [CrossRef]
25. Yuan, Y.; Lu, Z.; Yang, Z.; Jian, M.; Wu, L.; Li, Z.; Liu, X. Key frame extraction based on global motion statistics for team-sport videos. *Multimed. Syst.* **2021**, *28*, 387–401. [CrossRef]
26. Yang, H.; Wang, B.; Lin, S.; Wipf, D.; Guo, M.; Guo, B. Unsupervised extraction of video highlights via robust recurrent auto-encoders. In *Proceedings of the IEEE International Conference on Computer Vision*; IEEE: Santiago, Chile, 2015; pp. 4633–4641.
27. Mahasseni, B.; Lam, M.; Todorovic, S. Unsupervised video summarization with adversarial lstm networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Honolulu, HI, USA, 2017; pp. 202–211.
28. Yuan, L.; Tay, F.E.; Li, P.; Zhou, L.; Feng, J. Cycle-sum: Cycle-consistent adversarial lstm networks for unsupervised video summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI: Honolulu, HI, USA, 2019; Volume 33, pp. 9143–9150.
29. Yan, X.; Gilani, S.Z.; Feng, M.; Zhang, L.; Qin, H.; Mian, A. Self-supervised learning to detect key frames in videos. *Sensors* **2020**, *20*, 6941. [CrossRef]
30. Li, Y.; Luo, X.; Hou, S.; Li, C.; Yin, G. End-to-end Network Embedding Unsupervised Key Frame Extraction for Video-based Person Re-identification. In *11th International Conference on Information Science and Technology (ICIST)*; IEEE: Kopaonik, Serbia, 2021; pp. 404–410.
31. Elahi, G.M.E.; Yang, Y.H. Online learnable keyframe extraction in videos and its application with semantic word vector in action recognition. *Pattern Recognit.* **2022**, *122*, 108273. [CrossRef]
32. Ravi, S.; Larochelle, H. Optimization as a model for few-shot learning. In Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), Toulon, France, 24–26 April 2017.
33. Snell, J.; Swersky, K.; Zemel, R.S. Prototypical Networks for Few-Shot Learning. *arXiv* **2017**, arXiv:1703.05175.
34. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.H.; Hospedales, T.M. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Salt Lake City, UT, USA, 2018; pp. 1199–1208.
35. Oreshkin, B.; Rodríguez López, P.; Lacoste, A. Tadam: Task dependent adaptive metric for improved few-shot learning. *Adv. Neural Inf. Process. Syst.* **2018**, *31*.
36. Gidaris, S.; Bursuc, A.; Komodakis, N.; Pérez, P.; Cord, M. Boosting few-shot visual learning with self-supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: Seoul, Korea, 2019; pp. 8059–8068.
37. Hong, J.; Fang, P.; Li, W.; Zhang, T.; Simon, C.; Harandi, M.; Petersson, L. Reinforced attention for few-shot learning and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Nashville, TN, USA, 2021; pp. 913–923.
38. Wei, R.; Mahmood, A. Optimizing Few-Shot Learning Based on Variational Autoencoders. *Entropy* **2021**, *23*, 1390. [CrossRef] [PubMed]
39. Hsu, K.; Levine, S.; Finn, C. Unsupervised Learning via Meta-Learning. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
40. Ji, Z.; Zou, X.; Huang, T.; Wu, S. Unsupervised few-shot feature learning via self-supervised training. *Front. Comput. Neurosci.* **2020**, *14*. https://doi.org.10.3389/fncom.2020.00083. [CrossRef] [PubMed]
41. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]
42. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Las Vegas, NV, USA, 2016; pp. 770–778.

43. Mentzelopoulos, M.; Psarrou, A. Key-frame extraction algorithm using entropy difference. In *Proceedings of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval*; ACM: New York, NY, USA, 2004; pp. 39–45.
44. Nandini, H.M.; Chethan, H.K.; Rashmi, B.S. Shot based keyframe extraction using edge-LBP approach. *J. King Saud Univ. Comput. Inf. Sci.* 2020, *in press*. [CrossRef]
45. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; Morgan Kaufmann Publishers: Lake Tahoe, NV, USA, 2012; pp. 1097–1105.
46. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Salt Lake City, UT, USA, 2018; pp. 8697–8710.
47. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
48. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Boston, MA, USA, 2015; pp. 1–9.
49. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Honolulu, HI, USA, 2017; pp. 4700–4708.
50. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Las Vegas, NV, USA, 2016; pp. 2818–2826.
51. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Honolulu, HI, USA, 2017; pp. 1251–1258.
52. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*; AAAI: San Francisco, CA, USA, 2017.
53. van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.