A Transformer-based Late-Fusion Mechanism for Fine-Grained Object Recognition in Videos

Jannik Koch[1] Stefan Wolf[2,1] Jürgen Beyerer[1,2,3]

[1]Fraunhofer IOSB
Karlsruhe, Germany

[2]Vision and Future Lab
Karlsruhe Institute of Technology
Karlsruhe, Germany

[3]Fraunhofer Center
for Machine Learning
Munich, Germany

firstname.lastname@iosb.fraunhofer.de

Abstract

Fine-grained image classification is limited by only considering a single view while in many cases, like surveillance, a whole video exists which provides multiple perspectives. However, the potential of videos is mostly considered in the context of action recognition while fine-grained object recognition is rarely considered as an application for video classification. This leads to recent video classification architectures being inappropriate for the task of fine-grained object recognition. We propose a novel, Transformer-based late-fusion mechanism for fine-grained video classification. Our approach achieves superior results to both early-fusion mechanisms, like the Video Swin Transformer, and a simple consensus-based late-fusion baseline with a modern Swin Transformer backbone. Additionally, we achieve improved efficiency, as our results show a high increase in accuracy with only a slight increase in computational complexity. Code is available at: `https://github.com/wolfstefan/tlf`.

1. Introduction

Fine-grained classification is an important task in the context of surveillance since the identification of vehicles by licence plate is limited due to criminals often using stolen licence plates. Thus, fine-grained vehicle classification can be applied in a security context to identify vehicles by their make and model when an identification by licence plate fails. In real-world surveillance scenarios, a single image is limiting fine-grained vehicle classification since motion blur can render important classification features unrecognisable. This can be compensated by using videos for classification which are typically available anyway. Additionally, multiple views from different cameras can be exploited to increase the accuracy by using a video instead of a single image.

Figure 1: Example images from a video [12] which is part of the YouTube-Cars [35] dataset. Besides the advantage of multiple views, video classification enables the compensation of inappropriate images like the lower left one which shows drastic motion blur.

Video data has been successfully used as a source for various classification tasks. The most common example of this is the field of action recognition, where videos are advantageous because they provide important temporal information. However, the potential of using a frame sequence as opposed to a single image is not limited to the additional temporal component. As time progresses, a video usually provides different views of the objects in the scene, yielding additional features that can be exploited for classification tasks other than action recognition. Fine-grained object recognition is such a task that is likely to profit heavily from multiple views. But the availability of multiple frames is also helpful to compensate images inappropriate for classification because of *e.g.* blur. Nonetheless, only few works consider video classification for typical fine-grained object recognition tasks like vehicle model classification or bird species classification [1, 10, 24, 35]. This leads to state-of-the-art video classification models being tailored towards

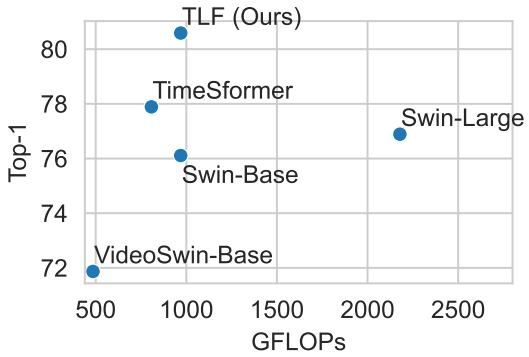

Figure 2: Comparing the computational complexity of our TLF architecture to Swin Transformer [17] models of different size with simple average fusion and the state-of-the-art video classification models Video Swin Transformer [18] and TimeSformer [3]. It shows the high efficiency of our approach.

action recognition and ignoring the unique challenges of fine-grained object recognition when processing videos.

The goal of fine-grained classification is to successfully differentiate a set of highly detailed classes. For example, in fine-grained vehicle classification Audi A5 Coupe 2012 could be such a class. In contrast, in regular coarse-grained classification, cars usually share a single class and have to be distinguished from *e.g*. humans. Due to this degree of class specificity in fine-grained classification, two classes might share the vast majority of features, leading to minor differences being the deciding factor.

Video data can increase the number of visible differences and thus, increase the classification accuracy. A major part of using video data is the fusion mechanism used to combine the input frames. Most state-of-the-art video classification architectures like the Video Swin Transformer [18] use an early-fusion approach which interrelates frames as they pass through the backbone as this proved to be advantageous for action recognition. Earlier approaches, like Temporal Segment Networks [31] use a simple late-fusion consensus mechanism. We pick the concept of late-fusion up again and combine it with a modern self-attention-based Transformer [30]. This results in our Transformer-based late-fusion mechanism (TLF).

In the following sections, we demonstrate superior results to both state-of-the-art early-fusion and strong baseline late-fusion models by applying our more sophisticated late-fusion approach. We achieve an improvement in accuracy without a significant increase in computational overhead, unlike the improvements resulting from a larger backbone network as can be seen in Figure 2.

Our main contributions are:

- proposing a sophisticated Transformer-based late-fusion mechanism that efficiently aggregates the features of multiple input images of a video to enhance the exploitation of the different views resulting in a significantly higher accuracy.
- showing the advantage of a sparse sampling strategy for fine-grained object recognition while this strategy is mostly considered outdated in video classification research.
- proving the effectiveness of video classification for fine-grained object recognition compared to single image classification.

2. Related work

In this section, we first discuss the existing literature in terms of video classification. Since most research in video classification is targeted towards action recognition, we summarize the literature focused on fine-grained video classification separately afterwards.

2.1. Video classification

Video classification is mostly researched in the context of action recognition which would be heavily limited by using single images. Thus, multiple datasets have been published for video-based action recognition [4, 9, 11, 14, 25, 27] and a large number of approaches has been proposed to optimize the accuracy on these datasets.

Two-stream architectures. As the temporal data can be conceptualized as another separate stream of information, two-stream architectures have been introduced as a viable option. Two-stream ConvNets [26] use both the RGB and optical flow of the input frames for separate classification tasks, the results of which are merged by a simple consensus mechanism. This late-fusion allows for a clean separation of two different backbone networks to extract the relevant features, but does not interrelate spatial and temporal information for the most part. If both domains are to be taken into account simultaneously, the backbone needs to directly work on a three-dimensional input.

Temporal Segment Networks [31] extend the two-stream ConvNets by employing a sparse sampling strategy that extracts multiple snippets of a video to acquire more information with each snippet containing a single RGB frame and a stack of optical flows.

3D convolutions. Since 2D convolutions are a common building block in classification architectures, convolutional architectures extending this concept along the temporal axis, like C3D [29], have followed accordingly. 3D convolutions and two-stream architectures are not mutually exclusive, leading to two-stream convolutional approaches like I3D [4].

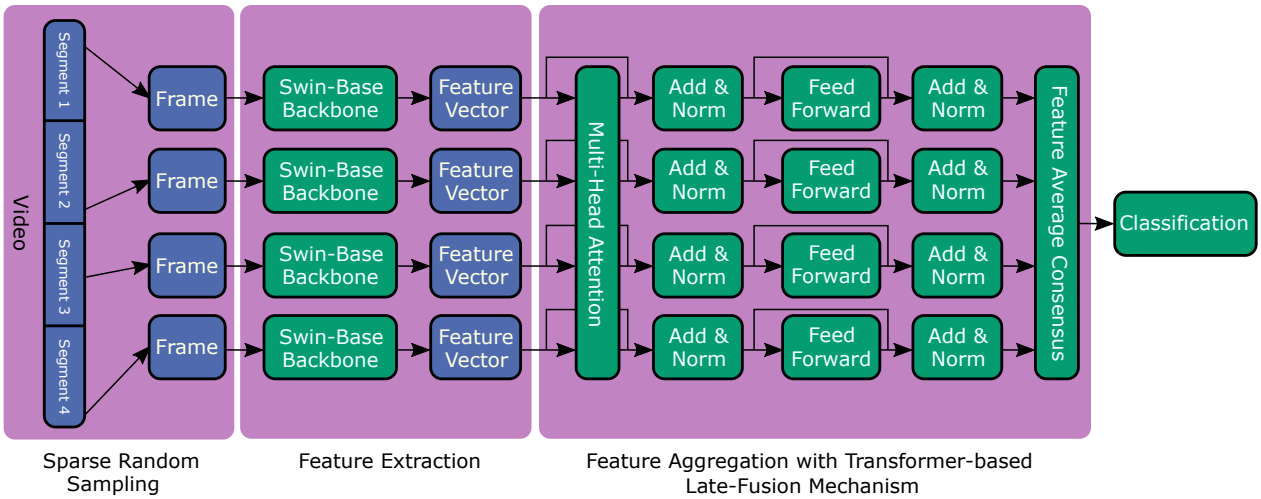

Figure 3: Schematic illustration of our Transformer-based late-fusion approach for fine-grained video classification.

Temporal Transformer. The recent trend of using self-attention for image recognition has also been picked up for video classification. These architectures use the Transformer mechanism to process sequences of images. ViViT [2] extends the ViT [8] architecture to enable the processing of image sequences. The authors propose multiple model variants including early-fusion and late-fusion methods. Neimark et al. [21] also propose a late-fusion Transformer architecture. However, early-fusion approaches have prevailed due to being advantageous for action recognition as shown by Arnab et al. [2]. While these late-fusion architectures are the most similar ones compared to our Transformer-based late-fusion mechanism, we show that fine-grained video classification has drastically different requirements leading to different design decisions. Video Swin Transformer [18] continues the trend of early-fusion architectures with the extension of the shifted windows mechanism of Swin Transformer [17] to the temporal dimension. The shifted windows reduce the computational complexity while ensuring inter-token information sharing. Bertasius et al. [3] propose TimeSformer as another Transformer-based video classification architecture that divides the spatial and temporal attention to reduce complexity and increase accuracy while using an early-fusion approach due to being tailored towards action recogintion.

2.2. Fine-grained classification

This section is divided into fine-grained classification based on images and videos. For the first, a large set of datasets [13, 28, 33] has been published motivating a variety of algorithmic approaches. In contrast, the field of fine-grained video classification is rather small. Most of the fine-grained video classification datasets and research works are about fine-grained action recognition [7, 14, 16, 23, 25] while only few datasets [1, 10, 24, 35] exist for our application of fine-grained object recogntion in videos limiting the reserch progress.

Fine-grained image classification. While fine-grained classification on single images can technically be realized using conventional image classification methods, specialized architectures have emerged to improve their results [15, 22, 34]. Initial models focused on the explicit identification of parts to distinguish the various classes, but with the advent of these deep learning approaches, identifying relevant features became both implicit and learnable.

Fine-grained video classification. In the context of fine-grained video classification, the research is significantly more narrow. Alsahafi et al. [1] use object detection to localize vehicles in videos and extract the relevant parts of the images. Afterwards, an imagewise CNN and a simple fusion mechanism are used for classification. Redundancy Reduction Attention [35] uses spatial and temporal attention to suppress redundant information in a video in an iterative manner.

3. Method

Our method is based on three parts which are illustrated in Figure 3. First, we extract frames from the input video by a sparse sampling strategy to cover the full range of the video. Afterwards we extract features of the frames with a modern Swin Transformer [17] backbone. As the last step, we apply a sophisticated Transformer-based late-fusion mechanism to derive a fine-grained classification score for each input video. Since fusion mechanisms implemented in the backbone might fail to find meaningful feature relationships early on, we postpone this operation to the

classification head. Late-fusion approaches like this usually rely on simple consensus mechanisms like averaging feature vectors, followed by a fully-connected, final classification layer. Temporal Segment Networks use this technique with great success [31], but have limited applicability in fine-grained classification due to their lack of attention across frame boundaries. Our approach prepends a Transformer encoder [30] to the consensus mechanism, which applies self-attention across all frames simultaneously, emphasizing important features. This self-attention provides an additional pathway to improving model accuracy in fine-grained classification, as correctly distinguishing closely-related classes might depend on very few features.

Sampling. In the first step of our approach, we apply a sparse sampling strategy that splits the video into a specific number of segments and selects a random frame from each segment. The set of frames is called *N*. Each resulting frame is augmented as described in Section 4. While the number of images is multiplied during inference due to augmentation, all preprocessed images are handled as a single set of input images *S*.

Sparse sampling is rarely used for video classification due to dense sampling being advantageous for action recognition which requires the extraction of short-term context. While Wang et al. [31] propose the sparse sampling strategy for action recognition by sampling the RGB frames sparsely across the video, they still employ some form of dense sampling by using stacks short-term optical flows in a two-stream architecture. However, fine-grained object recognition is profiting from using videos in a different way and thus, the widely applied dense sampling is inferior to a sparse sampling strategy as we show in our experiments.

Feature extraction. Each input image is fed separately into the Swin Transformer [17] backbone. The result is a feature vector with the shape of 1024x7x7 for each augmented image. Afterwards, each feature map is reduced to a feature vector of shape 1024x1x1 via adaptive average pooling. The total set of feature vectors of all augmentations from all frames is called *E*.

Self-attention. The self-attention mechanism central to our late-fusion mechanism is part of the Transformer architecture [30]. Specifically, self-attention is achieved via the multi-head attention block. Usually, this block receives different query *Q*, key *K* and value *V* matrices. In the case of self-attention, all these matrices are identical, meaning they will equally be set to the input features.

Each head learns its own set of linear transformations that are applied to the input data, followed by a scaled dot-product attention mechanism. All head outputs are eventually concatenated and linearly transformed to the desired output dimension. The scaled dot-product attention is implemented by multiplying *Q* and *K* and scaling the result in relation to their dimensionanilty, followed by a softmax. Scaling is required to prevent the dot-product from growing too large in magnitude.

The resulting matrix scores the importance of each input element with a value between 0 and 1. This attention matrix is then applied to *V* via matrix multiplication. Overall, attention can be expressed concisely via Equation 1.

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V \qquad (1)$$

Once the multi-head attention is defined, building a full Transformer encoder only requires two additional components. First, a residual connection has to be inserted, followed by an addition and normalization step that combines the residual data with the results. Second, a feed-forward network is appended to reduce the output dimension to the desired shape.

Transformer-based late-fusion. The feature vectors *E* are all passed through the Transformer encoder simultaneously. For our architecture, we use 8 heads within the multi-head attention model and 1024 for both the input features and the dimension of the feed-forward network. Hence, the resulting feature vectors still have the shape 1024x1x1. These feature vectors are averaged to provide a single 1024x1x1 feature vector to the classification stage.

Compared to a simple average fusion without the Transformer encoder, our Transformer-based late-fusion mechanism enables a more sophisticated aggregation. It can represent interdependencies between the views of images that can not be represented by a simple linear aggregation of the features.

Classification. Once the consensus has been applied, the resulting feature vector is fed into a dropout and a final fully-connected layer to classify the video. Afterwards, a softmax is applied to normalize the output scores.

4. Experiments

In the following paragraphs, the implementation, results and evaluation of our experiments will be discussed.

4.1. Settings

Optimization. The AdamW [20] optimizer with a Cosine Annealing [19] policy was consistently used during training. The learning rate was also kept consistent at a base value of 10^{-4} when combined with a batch size of 8. Due to VRAM limitations, this batch size had to be halved to 4 in some experiments. In these instances, the learning rate was also halved accordingly to $5 \cdot 10^{-5}$. All experiments use a weight decay of 10^{-2}. These values are based on the defaults used in mmaction2 [6].

Video decoding. The Decord [5] video loader used by mmaction2 provides two different modes of operation: *efficient* and *accurate*. Choosing the efficient mode reduces the time it takes to extract random frame samples, as Decord then utilizes a fast, inexact random seek algorithm that only returns Intra-Frames (or I-Frames). The drawback is the possibility of receiving the same frame twice when two samples are sufficiently close to each other. We chose to employ the efficient mode only during training, if samples are drawn in a sparse fashion. For dense sampling and during testing, accurate sampling was used. This yielded a compromise of lowered training times allowing possibly duplicated frames and preventing frame duplications during evaluations.

Sampling. Processing all images in a video is inappropriate for classification due to limited compute resources and redundancy of consecutive frames. Thus, a sampling strategy has to be applied to select a number of images that can be realistically handled and that are most appropriate for an accurate classification. In video classification, sparse and dense sampling are the common sampling strategies. In sparse sampling, the video is divided into k parts and from each part, a single image is selected randomly. In dense sampling, the video is also divided in k parts. However, from each part, a contiguous sequence of images with length l and stride s is chosen randomly. For experiments of dense sampling, we report the parameters as $l \times s \times k$.

Augmentations. During training, random horizontal flip is used per video and each sampled input image is augmented with a random crop. Afterwards, the frame is resized to 224x224 pixels. Each random crop has a random position in the frame, with all possible positions being equally probable. The dimension is calculated based on a given aspect ratio and the total area covered, both of which are chosen randomly within a given interval. In our case, the aspect ratio is chosen randomly in the interval $[\frac{3}{4}, \frac{4}{3}]$ and the area in $[0.08, 1]$, with the area being interpreted as a percentage of the total frame size. For testing, each frame is cropped 5 times: once in each corner and once in the center of the frame. This time, all crops have a static width and height of 224 pixels. Additionally, each augmentation is duplicated and flipped along the vertical axis, yielding 10 augmentations in total. Once an input frame has been augmented and resized to the input dimension of 224x224 pixels, the results are passed into the backbone network.

For Video Swin Transformer [18] and TimeSformer [3], we use a three crop strategy as intended by their authors. However, preliminary experiments have shown that the differences between three crop and ten crop are negligible.

4.2. Dataset

The YouTube-Cars dataset [35] is used to evaluate our architecture, since it provides video data with fine-grained labels. Additionally, the YouTube-Birds dataset provided by the same authors is used for additional validation of the model's efficacy. Experiments are done on the YouTube-Cars dataset if not mentioned otherwise. YouTube-Cars provides video data for 196 classes and YouTube-Birds for 200 classes, with the class selection being identical to Stanford Cars [13] and CUB-200-2011 [32], respectively. The full car dataset contains 10,238 videos for training and 4,855 videos for testing purposes while the bird dataset provides 12,666 training and 5,684 testing videos.

As YouTube is an inherently unreliable data source, video availability is never guaranteed. Thus, some of the videos of the datasets were not available anymore when we feteched the dataset. Hence, any comparison of our work to the results in the original paper is limited in its validity and should be considered tentative. However, both datasets were kept consistent during our experiments. While some videos were not available anymore, most of the data could still be fetched and no class had to be removed due to a lack of footage.

4.3. Comparison with state-of-the-art

To prove the effectiveness of TLF, we compare our approach against Swin Transformer [17] with a simple feature average consensus as a strong baseline model and the state-of-the-art video classification models Video Swin Transformer [18] and TimeSformer [3]. While we also include published results on the YouTube-Cars dataset [35], the results are not directly comparable due to some videos not being available anymore as described in Section 4.2. We used the best of all evaluated sampling strategies and number of samples for each model. The impact of the sampling strategy is described in Section 4.4. The results of the comparison with the state-of-the-art are shown in Table 1 and indicate an advantage of the baseline Swin Transformer model with a feature averaging fusion over the Video Swin Transformer which is a state-of-the-art video classification model. The Video Swin Transformer only performs well for tasks requiring an analysis of short sequences like action recognition but not for tasks covering long range correspondences in videos. This highlights the different requirements of action recognition as the prime task for video classification research and fine-grained object recognition which has not received the same attention in research yet. The TimeSformer model can outperform our baseline slightly since it can make appropriate use of sparse sampling. However, due to the Transformer-based fusion being applied in an early-fusion manner, the architecture still cannot make use of the Transformer to ist full advantage. In comparison, our simple TLF mechanism combined with a Swin-Base backbone

Architecture	Top-1	Top-5	#parameters ($\mathbf{10^6}$)	FLOPs ($\mathbf{10^9}$)
Swin Transformer Base [17]	76.1	93.7	86.9	969.0
Swin Transformer Large [17]	76.9	94.5	195.3	2178.5
VideoSwinTransformer Base [18]	71.9	90.6	87.8	485.1
TimeSformer [3]	77.9	94.6	121.5	807.2
Transformer-based Late-Fusion (Ours)	**80.6**	**96.0**	93.3	969.1
Inflated 3D Convolutional Neural Network (I3D)*	40.9			
Batch-Normalized-Inception (BN-Inception)*	62.0			
Temporal Segment Network (TSN)*	74.3			
Redundancy Reduced Attention (RRA)*	77.6			

* Original results by the authors of the YouTube-Cars dataset [35]. Results are not directly comparable since not all videos are available anymore.

Table 1: Results of different classification architectures on the YouTube-Cars dataset [35]. The state-of-the-art video classification Video Swin Transformer performs poorly for fine-grained vehicle recognition due to the early-fusion design optimized for action recognition. A modern single image backbone with a simple late-fusion average consensus achieves better results. In comparison, our Transformer-based late-fusion mechanism shows a significantly higher accuracy.

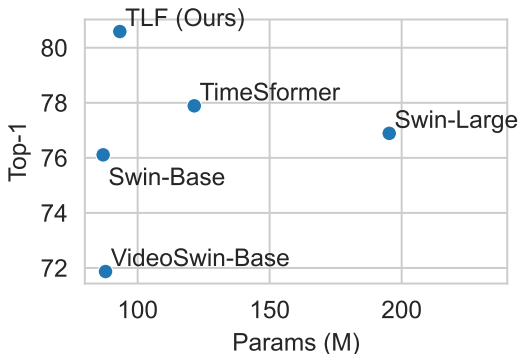

Figure 4: A comparison of our TLF architecture to the Swin Transformer [17], the Video Swin Transformer [18] and the TimeSformer [3] models. Our model achieves a superior top-1 accuracy to the other models with only a slight increase in parameters.

shows a significant advantage over the baseline and TimeSformer and even more over the Video Swin Transformer. This is achieved by using a model design which is targeted towards video-based fine-grained object recognition. The advantage is particularly remarkable considering the small increase in computational complexity compared to the increase in accuracy as shown in Figure 2. This also holds true when considering the number of parameters which is shown in Figure 4. All FLOPs and number of parameters are reported for the case of 64 input samples and ignoring augmentations for a fair comparison.

4.4. Ablation studies

In this section, the impact of the improvements and design choices is evaluated.

Architecture and sampling. We found the sampling of the video to be a deciding factor for the accuracy of the classification. In Table 2, sparse and dense sampling with different number of sampled images during training and testing are compared for different models. For the Swin-Base model without TLF and with sparse sampling, we see a significant increase in accuracy with the number of testing samples increased from 8 to 32 and a slight increase when the number of training samles is increased from 8 to 16. Increasing the number of testing samples further to 64 does not lead to another performance improvment. This increase can be explained by the higher number of perspectives available with a higher frame number. Additionally, more samples enable the compensation of inappropriate samples with other samples.

Using dense sampling with the Swin-Base model leads to a drastic drop in accuracy due to the lack of variance of continuously sampled images. Increasing the number of sampled clips to four during testing for a total number of 64 frames, the accuracy increases again but is still lower than using 8 frames with a sparse sampling strategy. In contrast to these results, VideoSwin-Base does not perform well with sparse sampling showing a large drop compared to the baseline model. Dense sampling with an increased number of clips closes the gap but the performance is still worse than the baseline. VideoSwin-Base performs an early fusion which requires a high similarity of the images. This is only sensible with a dense sampling strategy explaining the lower accuracy with sparse sampling. However, for fine-grained object recognition, a dense sampling strategy is not appropriate due to a high variety of perspectives being the most important factor for efficiently exploiting videos.

TimeSformer shows a low accuracy with its default dense sampling strategy. However, it can profit signifi-

Model	Sampling Strategy	Training Samples	Testing Samples	Top-1	Top-5
Swin-Base	Sparse	8	8	73.2	92.1
Swin-Base	Sparse	8	32	75.6	**94.0**
Swin-Base	Sparse	16	32	**76.1**	93.9
Swin-Base	Sparse	16	64	**76.1**	93.7
Swin-Base	Dense	16×2×1	16×2×1	55.0	78.0
Swin-Base	Dense	16×2×1	16×2×4	72.5	92.2
VideoSwin-Base	Sparse	16	64	70.0	**91.2**
VideoSwin-Base	Dense	16×2×1	16×2×1	46.5	71.6
VideoSwin-Base	Dense	16×2×1	16×2×4	**71.9**	90.6
TimeSformer	Sparse	16	32	76.7	94.0
TimeSformer	Sparse	16	64	**77.9**	**94.6**
TimeSformer	Dense	8×32×1	8×32×1	58.3	80.7
TimeSformer	Dense	8×32×1	8×32×4	45.1	69.5
Swin-Base + TLF (Ours)	Sparse	16	32	80.5	**96.0**
Swin-Base + TLF (Ours)	Sparse	16	64	**80.6**	**96.0**

Table 2: Comparison of sample type and sample sizes during training and testing. Swin-Base with a simple average feature fusion performs best with sparse sampling and a high number of samples during testing while the number of samples during training has only a small impact. Dense sampling performs worse for Swin-Base while it is advantageous for VideoSwin-Base which relies on a high similarity of images in a single clip due to its early-fusion approach. Our TLF performs best since it combines the strategy of a sophisticated fusion and a late-fusion.

cantly from using a sparse sampling strategy and with using the sparse sampling, it is slightly superior compared to our baseline.

Since our TLF approach performs a late-fusion, we apply sparse sampling for it. Even with only 32 images during testing, it outperforms all other evaluated models by a significant margin. Increasing the number of testing samples from 32 to 64 shows a slight increase in terms of accuracy.

Positional encoding. Transformer architectures usually apply a positional encoding to provide the information about the sequence of the inputs to the model. Since the original proposal of the transformer architecture [30], it is the default setting to use a positional encoding. Thus, we also evaluate the application of a positional encoding. For the experiment, we use the original fixed sine-based positional encoding as proposed by Vaswani et al. [30]. The results are shown in Table 3. The positional encoding is a significant disadvantage for this task with the architecture without a positional encoding achieving a higher accuracy. Thus, we drop the positional encoding for our TLF approach.

The reason for the negative impact of the positional encoding is likely an overfitting of the network when it is able to identify the ordering of the frames and thus expects the same order during inference. Moreover, for fine-grained object recognition the ordering of the frames is rarely important since most frames are of different scenes anyway due to cuts occurring in the videos.

Positional encoding	Top-1	Top-5
Yes	76.2	93.6
No	**80.5**	**96.0**

Table 3: Evaluation of positional encoding. We applied a fixed sine-based positional encoding as it is common for transformer architectures. However, a positional encoding has shown a significant drop in accuracy for our use case.

4.5. Effectiveness of video classification

In Table 4, we compare the use of videos to the use of single images for fine-grained classification of cars. For the comparison, we use a Swin-Base model with an average consensus fusion for the video classification and a plain Swin-Base model for the image classification. In both cases, sparse sampling with 8 frames in training and 32 frames in testing is used. For the per-frame evaluation, each frame is evaluated individually and the average accuracy over all frames is calculated. Since the number of sampled frames per video is constant, the results are comparable to the per-video evaluation results. As can be seen, the use of per-video evaluation provides a drastic increase in accuracy. This shows the advantage of video classification compared to single image classification for fine-grained object recognition and should motivate future research in this direction. The expressiveness of these results might be limited due to

Evaluation	Top-1	Top-5
Per-frame	64.9	86.5
Per-video	**73.9**	**92.6**

Table 4: Comparison of a per-frame evaluation and a per-video evaluation. In case of the per-frame evaluation, the sampling is unchanged but no average is calculated over the images. Instead, each image is evaluated separately. For the per-video evaluation, a simple feature average fusion is used. The comparison shows that using videos has a clear advantage over using only frames since the combination of multiple frames provides additional information useful for classification.

Architecture	Top-1	Top-5
Swin Transformer Base [17]	76.9	91.1
VideoSwinTransformer [18]	72.8	88.2
TimeSformer [3]	77.6	90.9
TLF (Ours)	**78.9**	**91.3**
I3D*	40.7	
BN-Inception*	60.1	
TSN*	72.4	
RRA*	73.2	

* Original results by the authors of the YouTube-Birds dataset [35]. Results are not directly comparable since not all videos are available anymore.

Table 5: Results of different classification architectures on the YouTube-Birds dataset [35]. Our transformer-based late fusion mechanism outperforms the baseline by a significant margin due to the more sophisticated fusion of frames.

some frames showing irrelevant information and being useless for classification. However, for practical applications like surveillance, a single frame can also be suboptimal for object recognition due to *e.g*. being blurred. In this case, video classification can compensate single blurred images by ignoring the frame and using information from other frames.

4.6. Results on YouTube-Birds

To show the general effectiveness of our approach for fine-grained video classification, we compare our TLF to a strong baseline on YouTube-Birds [35]. The results are shown in Table 5. We sample 64 frames in total per video with the best sampling strategy chosen per model. Similar to the results on YouTube-Cars, our model shows a significant increase in terms of classification accuracy compared to the baseline and state-of-the-art models.

4.7. Discussion of results

While our approach outperforms the competitive architectures presented, the influential effect of how the videos are sampled deserves to be emphasized again. Switching between dense and sparse sampling can significantly affect the final accuracy of the model, with no sampling strategy being blatantly superior. As an example, the Video Swin Transformer thrives on dense sampling, while the Swin Transformer performs better when sparsely sampling the video. This indicates that the model architecture might dictate the sampling strategy, with dense sampling being preferably used with early-fusion and sparse sampling with late-fusion approaches. Nonetheless, overall the results show that a sparse sampling strategy with a late-fusion approach is superior. Furthermore, models tend to benefit from higher sample counts, both during training and testing. A fair comparison therefore requires two models to receive an identical amount of frames. However, while the frame count might be identical, the frames themselves might not be. This can cause issues due to the high variance in frame quality, which we define as the volume of useful information a frame provides. Additionally, frames might have a low information content regardless of the sampling mode. Exemplary causes of this in the YouTube-Cars dataset are transitions within the video, scenes showing the car interior or frames where the car is occluded by people or other cars. These issues are mitigated in some cases by sampling a sufficient amount of frames, but due to the inherent randomness of the sampling and the varying information density of the video data, this can not be guaranteed.

5. Conclusion

We propose a sophisticated late-fusion approach for fine-grained object recognition using video data. By adding self-attention through a transformer encoder, a simple average consensus mechanism can be extended to achieve results superior to both a state-of-the-art video classification architecture and the basic consensus mechanism with a larger backbone network. The transformer encoder applied in the late stages of the network enables the fusion of semantically high-level features and thus, better exploits the multiple views offered by video data. Since the presented mechanism comes with low additional cost in terms of FLOPs and parameters, it is more applicable in a real-time setting than a larger model with comparable accuracy.

Finally, we show that proper sampling is a central factor in classification accuracy, both in terms of sample count and sample distribution across the input video. Making the sampling strategy more intelligent and focused on yielding frames containing useful information instead of relying on randomness is an important area of future work that could drastically improve results.

References

[1] Yousef Alsahafi, Daniel Lemmond, Jonathan Ventura, and Terrance Boult. Carvideos: A novel dataset for fine-grained car classification in videos. In *16th International Conference on Information Technology-New Generations (ITNG 2019)*, 2019.

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *38th International Conference on Machine Learning*, 2021.

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] Decord Contributors. Decord. `https://github.com/dmlc/decord`, 2022.

[6] MMAction2 Contributors. Openmmlab's next generation video understanding toolbox and benchmark. `https://github.com/open-mmlab/mmaction2`, 2020.

[7] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018.

[8] 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. In *International Conference on Learning Representations*, 2021.

[9] Haodong Duan, Yue Zhao, Yuanjun Xiong, Wentao Liu, and Dahua Lin. Omni-sourced webly-supervised learning for video recognition. In *European Conference on Computer Vision (ECCV)*, 2020.

[10] ZongYuan Ge, Chris McCool, Conrad Sanderson, Peng Wang, Lingqiao Liu, Ian Reid, and Peter Corke. Exploiting temporal information for dcnn-based fine-grained object classification. In *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2016.

[11] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[12] Kline Innovation. Audi r8 gt, valvetronic race exhaust by kline innovation. `https://www.youtube.com/watch?v=VDDQhOaj1ss`, 2016. Accessed: 2022-11-17. Licence: CC BY 3.0 `https://creativecommons.org/licenses/by/3.0/`.

[13] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, 2013.

[14] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *European Conference on Computer Vision (ECCV)*, 2018.

[15] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[16] Shenglan Liu, Xiang Liu, Gao Huang, Hong Qiao, Lianyu Hu, Dong Jiang, Aibin Zhang, Yang Liu, and Ge Guo. Fsd-10: A fine-grained classification dataset for figure skating. *Neurocomputing*, 413:360–367, 2020.

[17] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[18] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[19] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017.

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[21] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2021.

[22] Yuxin Peng, Xiangteng He, and Junjie Zhao. Object-part attention model for fine-grained image classification. *IEEE Transactions on Image Processing*, 27(3):1487–1500, 2017.

[23] AJ Piergiovanni and Michael S. Ryoo. Fine-grained activity recognition in baseball videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018.

[24] Tomoaki Saito, Asako Kanezaki, and Tatsuya Harada. Ibc127: Video dataset for fine-grained bird classification. In *2016 IEEE International Conference on Multimedia and Expo (ICME)*, 2016.

[25] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[26] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, 2014.

[27] K. Soomro, A. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *ArXiv*, abs/1212.0402, 2012.

[28] Faezeh Tafazzoli, Hichem Frigui, and Keishin Nishiyama. A large and diverse dataset for improved vehicle make and

model recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (2017) Workshops*, 2017.

[29] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[31] 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. In *European Conference on Computer Vision (ECCV)*, 2016.

[32] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. *California Institute of Technology. CNS-TR-2010-001*, 2010.

[33] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[34] Xiaopeng Zhang, Hongkai Xiong, Wengang Zhou, Weiyao Lin, and Qi Tian. Picking deep filter responses for fine-grained image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[35] Chen Zhu, Xiao Tan, Feng Zhou, Xiao Liu, Kaiyu Yue, Errui Ding, and Yi Ma. Fine-Grained Video Categorization with Redundancy Reduction Attention. In *European Conference on Computer Vision (ECCV)*, 2018.