VIDEOEPITOMA: EFFICIENT RECOGNITION OF LONG-RANGE ACTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

CNNs are widely successful in recognizing human actions in videos, albeit with a great cost of computation. This cost is significantly higher in the case of long-range actions, where a video can span up to a few minutes, on average. The goal of this paper is to reduce the computational cost of these CNNs, without sacrificing their performance. We propose VideoEpitoma, a neural network architecture comprising two modules: a timestamp selector and a video classifier. Given a long-range video of thousands of timesteps, the selector learns to choose only a few but most representative timesteps for the video. This selector resides on top of a lightweight CNN such as MobileNet and uses a novel gating module to take a binary decision: consider or discard a video timestep. This decision is conditioned on both the timestep-level feature and the video-level consensus. A heavyweight CNN model such as I3D takes the selected frames as input and performs video classification. Using off-the-shelf video classifiers, VideoEpitoma reduces the computation by up to 50% without compromising the accuracy. In addition, we show that if trained end-to-end, the selector learns to make better choices to the benefit of the classifier, despite the selector and the classifier residing on two different CNNs. Finally, we report state-of-the-art results on two datasets for long-range action recognition: Charades and Breakfast Actions, with much-reduced computation. In particular, we match the accuracy of I3D by using less than half of the computation.

1 INTRODUCTION

A human can skim through a minute-long video in just a few seconds, and still grasp its underlying story (Szelag et al., 2004). This extreme efficiency of the human visual and temporal information processing beggars belief. The unmatched trade-off between efficiency and accuracy can be attributed to visual attention (Szelag et al., 2004) – one of the hallmarks of the human cognitive abilities. This raises the question: can we build an efficient, yet effective, neural model to recognize minutes-long actions in videos?

A possible solution is building efficient neural networks, which have a demonstrated record of success in the efficient recognition of images (Howard et al., 2017). Such models have been successful for recognizing short-range actions in datasets such as HMDB (Kuehne et al., 2011) and UCF-101 (Soomro et al., 2012), where analysis of only a few frames would suffice (Schindler & Van Gool, 2008). In contrast, a long-range action can take up to a few minutes to unfold (Hussein et al., 2019a). Current methods fully process the long-range action video to successfully recognize it. Thus, for long-range actions, the major computational bottleneck is the sheer number of video frames to be processed.

Another potential solution is attention. Not only it is biologically plausible, but also it is used in a wide spectrum of computer vision tasks, such as image classification (Wang et al., 2017), semantic segmentation (Oktay et al., 2018), action recognition (Wang et al., 2018) and temporal localization (Nguyen et al., 2018). Attention has also been applied to language understanding (Lin et al., 2017) and graph modeling (Veličković et al., 2017). Most of these methods use soft-attention, where the insignificant visual signals are least attended to. However, such signals are still fully processed by the neural network and hence no reduction on the computation cost is obtained.

Neural gating is a more conceivable choice to realize the efficiency, by completely dropping the insignificant visual signals. Recently, there has been a notable success in making neural gating differentiable (Maddison et al., 2016). Neural gating is applied to conditional learning, and is used to gate network layers (Veit & Belongie, 2018), convolutional channels (Bejnordi et al., 2019), and more (Shetty et al., 2017). That begs the question: can neural gating help in reducing the computational cost of recognizing minutes-long actions? That is to say, can we learn a gating mechanism to consider or discard video frames, conditioned on their video?

Motivated by the aforementioned questions, we propose VideoEpitoma, a two-stage neural network for efficient classification of long-range actions without compromising the performance. The first stage is the timestep selector, in which, many timesteps of a long-range action are efficiently represented by lightweight CNN, such as MobileNet (Howard et al., 2017; Sandler et al., 2018; Howard et al., 2019). Then, a novel gating module learns to select only the most significant timesteps – practically achieving the *epitoma* (Latin for *summary*) of this video. In the second stage, a heavyweight CNN, such as I3D (Carreira & Zisserman, 2017), is used to effectively represent only the selected timesteps, followed by temporal modeling for the video-level recognition.

This paper contributes the followings: *i.* VideoEpitoma, a neural network model for efficient recognition of long-range actions. The proposed model uses a novel gating module for timestep selection, conditioned on both the input frame and its context. *ii.* Off the shelf, our timestamp selector benefits video classification models and yields signification reduction in computation costs. We also show that if trained end-to-end, the timestep selector learns better gating mechanism to the benefit of the video classifier. *iii.* We present state-of-the-art results on two long-range action recognition benchmarks: Charades (Sigurdsson et al., 2016) and Breakfast Actions (Kuehne et al., 2014) with significant reductions in the computational costs.

2 Related Work

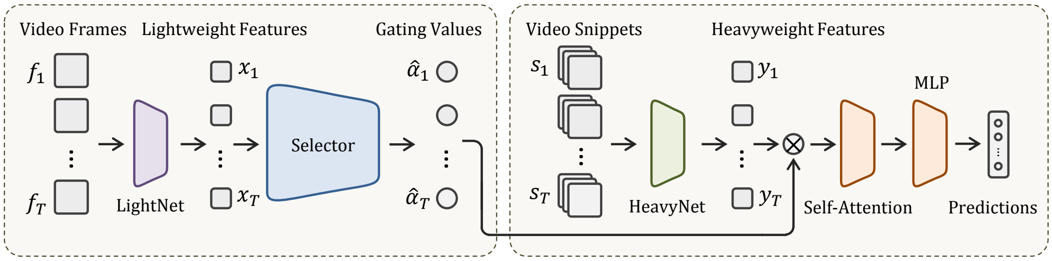
Efficient Architectures. CNNs are the go-to solution when it comes to video classification. Thus, one prospective of reducing the computation of video recognition is to build efficient CNNs. Methods for pruning least important weights (Hassibi et al., 1993; Han et al., 2015) or filters (Li et al., 2016) were previously proposed. Careful design choices result in very efficient 2D CNNs such as MobileNet (Howard et al., 2019) and ShuffleNet (Zhang et al., 2018). These 2D CNNs are extended to their 3D counterparts (ShuffleNet-3D and MobileNet-3D byKöpüklü et al. (2019)) to learn spatio-temporal concepts for video classification. Neural architecture search (Zoph & Le, 2016) is used to find the lightweight NasNet-Mobile (Zoph et al., 2018).

Long-range Actions Short-range actions in datasets such as Kinetics (Kay et al., 2017) and UCF-101 (Soomro et al., 2012) have average length of 10 seconds. They can be practically classified with CNNs using as little as 10 frames per video (Wang et al., 2016), and in some cases, even 1 frame would suffice (Schindler & Van Gool, 2008). Therefore, building efficient CNNs is a plausible choice to reduce computational cost of recognizing them. However, long-range videos (e.g. Charades (Sigurdsson et al., 2016) and Breakfast Actions (Kuehne et al., 2014)) can take up to 5 minutes to unfold. Thus, requiring as many as a thousand frames (Hussein et al., 2019a;b) to be correctly classified. As such, analyzing all the frames using efficient CNNs can still be computationally expensive. In contrast, having a mechanism to select the most relevant frames can boost the efficiency Bhardwaj et al. (2019). Therefore, this paper focuses on reducing the number of video frames needed for action recognition. Nevertheless, our work is orthogonal to prior works that focus on development of efficient CNN for action recognition.

Conditional Computing. Another solution to reduce the computation is to dynamically route the compute graph of a neural network. The assumption is that not all input signals require the same amount of computation – some are complicated while others are seemingly easy. Thanks to categorical reparametarization (Jang et al., 2016), it becomes possible to discretize a continuous distribution, and effectively learn binary gating. In (Veit & Belongie, 2018), a dynamical graph is build by gating the layers of a typical CNN. While in (Chen et al., 2019; Bejnordi et al., 2019), the gating is achieved on the level of convolutional channels. In the same vien, GaterNet (Chen et al., 2019) proposes a separate gater network to learn binary gates for the backbone network. Differently, this paper focuses on gating the video frames themselves, to realize efficiency.

Sampling of Video Frames. Several works discuss frame sampling for short-range videos. SC-Sampler (Korbar et al., 2019) learns a ranking score using trimmed *v.s.* untrimmed video segments. Bhardwaj et al. (2019) proposes a student-teacher model for trimmed video classification. In (Yeung et al., 2016), an agent is trained with reinforcement to learn where to look next. However, frame sampling for long-range actions is fundamentally different from that of short-range. Unlike short-range actions, in long-range actions, usually a much smaller proportion of timesteps are crucial for classification. As a result, this paper focuses on frame selection for solely long-range actions, and it does not require any video-level annotation other than the video category itself.

3 Method



(a) Timestep Selector (b) Video Classifier

Figure 1: Overview of the proposed model, VideoEpitoma, with two stages. The first stage is the Timestep Selector, left. Based on a lightweight CNN, *LightNet*, it learns to select the most relevant timesteps for classifying the video. This selection is conditioned on both the features of timestep and its context. The second stage is the video classifier, right. It depends on heavyweight CNN, *HeavyNet*, to effectively represent only timesteps selected in the previous stage. Then it temporally models these selected timesteps to arrive at the video-level feature, which is then classified.

Model Overview. VideoEpitoma consists of two stages: Timestep Selector and Video Classifier, see figure 1. The first stage is the Timestep Selector and consists of a lightweight CNN, *LightNet*, followed by a novel gating module, see figure 2. The purpose of this module is timestep gating, i.e. to take binary decision of considering or discarding each video timestep, based on how relevant it is to the video itself. The second stage is the video classifier. Its main purpose is to learn deep and discriminatory video-level representations for maximum classification accuracy. Thus, it resides on top of a heavyweight CNN, *HeavyNet*, followed by an off-the-shelf temporal layer for video-level representation, and a Multi-Layer Perceptron (MLP) for classification. Only the timesteps chosen by the first stage, *i.e.* the Timestep Selector, are considered by the second stage, *i.e.* the video classifier.

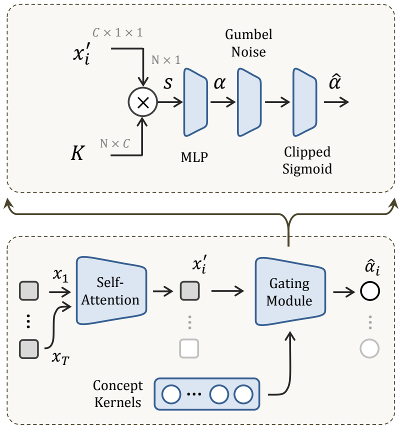


Figure 2: Bottom, the Timestep Selector learns concept kernels to represent the dominant visual concepts across the videos. Top, the gating module learns to select only a few timesteps according to their importance to the current video.

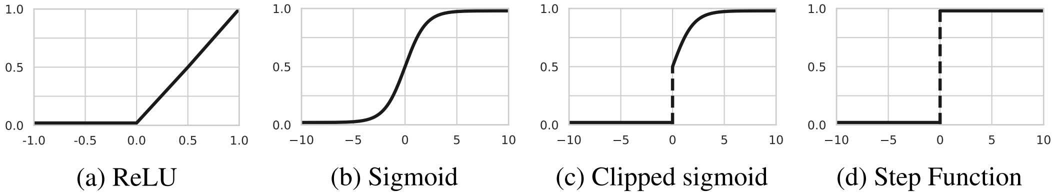
The Timestep Selector. Conceptually speaking, a long-range action consists of few yet dominant and discriminative visual concepts, based on which, the video can be recognized (Hussein et al., 2019b;a). Take for example "Making Pancake". One can easily discriminate it by observing its dominant evidences "Pancake", "Eggs", "Pan", and "Stove". These evidences can be thought of *latent* concepts. To represent these concepts, we opt for learning a set of N concept kernels $K = \{k_1, k_2, ...k_N\}$. K are randomly initialized and are part of the network parameters and learned during the training of the selector. Our concept kernels K are reminiscent of the nodes in VideoGraph (Hussein et al., 2019b) or the centroids in ActionVLAD (Girdhar et al., 2017).

Once these concepts are learned, it becomes easier to efficiently summarize a long-range action. We transverse through a video of thousands timesteps and decide which of them to consider and which to discard, based on the similarity between the features of these timesteps and that of the latent concepts. Our assumption is that a lightweight representation of each timestep is sufficient for taking this decision. Thus, the selector depends on an efficient LightNet to represent these timesteps. Given a long-range video v of T timestep, each is represented as a feature $x_i \in \mathbb{R}^{C \times H \times W}$ using the LightNet, where C is the convolutional channels, H, W are the channel height and width, respectively.

The Gating Module. The purpose of the gating module is to select the video timsteps, see figure 2 top. We start by comparing how relevant each timstep feature x_i is to all of the concept kernels $K \in \mathbb{R}^{N \times C}$ using a dot product. The result is the similarity scores $s_i = K \cdot x_i, s \in \mathbb{R}^{N \times 1}$, representing how relevant a timestep is to each of these concept kernels. Then we model the correlation between these similarity scores s_i with a two-layer MLP with a single neuron in the output layer, denoted as α. Next, we need to convert the continuous variable α to a binary variable, such that it represents the decision of the gating module. For this, we make use of (Jang et al., 2016) to discretize a continuous variable. Following the gating mechanism of (Bejnordi et al., 2019), we add gumbel noise to α and follow with *sigmoid* activation and binary thresholding, arriving at the activated gating value $\hat{\alpha}$. Then, each timestep feature x_i is multiplied by $\hat{\alpha}$, to either select or discard it.

A problem with the aforementioned gating mechanism is that during the feedforward, the classifier does not know which of the selected timesteps is more relevant than the other. As a remedy, we propose a different gating mechanism, see figure 2, top. First, a `sigmoid` non-linearity is applied to the gating value α to limit its lower- and upper-bound, $\hat{\alpha} = \texttt{sigmoid}(\alpha)$. Then, to achieve gating, we clip $\hat{\alpha}$ below threshold 0.5. This modified activation function `clipped_sigmoid` fits perfectly to the purpose of timestep gating due to 3 desirable properties, see figure 3. *i.* Being a relaxation for the step-function makes it differentiable. *ii.* Retaining the `sigmoid` value above the threshold means that the classifier gets the chance to know, out of the selected timesteps, which is relatively more important than the other. *iii.* Unlike `ReLU`, the `sigmoid` activation is upper-bounded by 1, thus preventing a single timestep from dominating the others by being multiplied by unbounded gating value $\hat{\alpha}$.

Context Conditional Gating. Up till now, the selector learns to gate each timestep regardless of its context, i.e. the video itself. To achieve conditional gating, where both the timestep and its context affect the gating mechanism, we opt for a temporal modeling layer, self-attention (Wang et al., 2018), before the gating module, See figure 2, bottom. This temporal layer learns to correlate each timestep with all the others in the video before gating.



(a) ReLU (b) Sigmoid (c) Clipped sigmoid (d) Step Function

Figure 3: For selecting timesteps during training, the gating module uses `gated-sigmoid` as the activation for the gating value α. It has some desirable properties. *i.* Unlike ReLU, having upper bound does not allow a timestep feature to dominate others. *ii.* Unlike `sigmoid`, being clipped allows the network to discard insignificant timesteps, i.e. those with gating values $\alpha < 0.5$. In test time, we replace the `gated-sigmoid` with `step-function` for binary gating of timesteps.

Sparse Selection. The last component of the selector is to enforce sparsity on timestep selection, *i.e.* choose as few timesteps as possible, yet retain the classification accuracy. Loosely speaking, the selector can simply cheat by predicting gating values α just higher than the threshold 0.5, resulting in all gates opened and all timesteps selected. The selector has a natural tendency to such a behaviour, as the only loss used so far is that of classification. And the more timesteps used by the classifier, the better the classification accuracy. To prevent such a behaviour, we apply L_0 (Louizos et al., 2017) regularization to the gating values $\hat{\alpha}$ to enforce sparsity on the selected timesteps. We note that the Sparsity regularization is necessary for a properly functioning gating mechanism.

The Video Classifier The assumption of VideoEpitoma is that having efficiently selected the most crucial timesteps from the video using the LightNet and the selector, one can opt for a much more powerful HeavyNet to effectively classify the video. Thus, the second stage of VideoEpitoma is the video classifier, see figure 1. It takes as input only the subset T' of timesteps chosen by the selector, $T' \ll T$. Each timestep is represented as a feature y_i using HeavyNet. Following feature extraction, we use one layer of self-attention for temporal modeling to obtain a video-level representation, followed by a two-layer MLP for the final classification.

3.1 Model Implementation

Before training VideoEpitoma, all the CNN used, as LightNet and HeavyNet, are fine-tuned first on the videos of the dataset in hand. VideoEpitoma is trained with batch size 32 and for 100 epochs. We use Adam with learning rate $1e$-3 and epsilon $1e$-4. We use PyTorch and TensorFlow for our implementation. Our choice for the LightNet is MobileNetv3. As for the HeavyNet, we experiment with I3D (Carreira & Zisserman, 2017), ShuffleNet3D and ResNet2D (He et al., 2016) (the 50-layer version). Worth mentioning that in the gating module, and during the training phase, we use gumbel noise and `clipped sigmoid` to get the activated gating value $\hat{\alpha}$, see figure 2. In the test phase, we don't use gumbel noise, and we use `step-function`, to get a binary gating value.

4 Experiments

4.1 Datasets

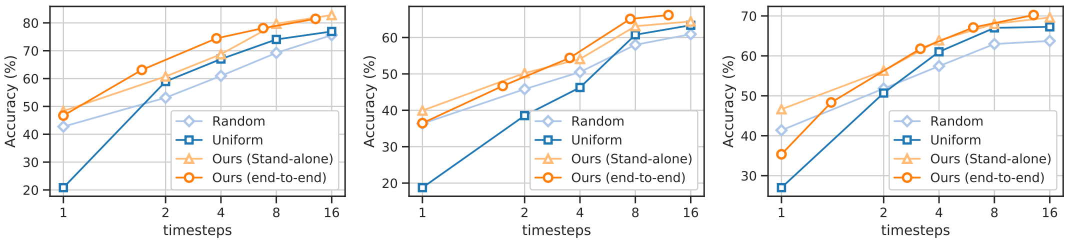
Breakfast Actions Breakfast Actions is a dataset for long-range actions, depicting cooking activities. All in all, it contains 1712 videos, divided into 1357 and 335 for training and testing, respectively. The task is video recognition into 10 classes of making different breakfasts. Added to the video-level annotation, we are given temporal annotations of 48 one-actions. In our experiments, we only use the video-level annotation, and we do not use the temporal annotation of the one-actions. The videos are long-range, with the average length of 2.3 minutes per video. Which makes it ideal for testing the efficiency of recognizing long-range actions. The evaluation method is the accuracy.

Charades Charades is a widely used benchmark for human action recognition. It is a diverse dataset with 157 action classes in total. The task is mult-label recognition, where each video is assigned to one or more action class. It is divided into 8k, 1.2k and 2k videos for training, validation and test splits, respectively, covering 67 hours. On average, each video spans 30 seconds, and is labeled with 6 and 9 actions for training and test splits, respectively. Thus, Charades meets the criteria of long-range actions. We use Mean Average Precision (mAP) for evaluation.

4.2 Stand-Alone Timestep Selector

One might raise an important question – will a Timestep Selector based on LightNet features benefit a classifier based on HeavyNet features, given the differences between the feature spaces of LightNet and HeavyNet? To answer this question, we construct an experiment of two steps on Breakfast. The first step is training a stand-alone selector. For this, we train VideoEpitoma to classify the videos of Breakfast, where we choose MobileNet for both LightNet and HeavyNet. During training, we randomly sample $T = 32$ timesteps from each video. Since MobileNet is a 2D CNN, a timestep here is practically a video frame. With the help of the L_0 regularization, the selector achieves sparse selection of timesteps, by as little as $T = 16$ without degrading the classification performance. The second step is testing how will the selector benefit off-the-shelf CNN classifiers. For this, we use different CNN classifiers, previously fine-tuned on Breakfast: I3D, ShuffleNet3D and ResNet2D. Then, we measure their performance using sampled $T \in \{1, 2, 4, 8, 16\}$ timesteps from each video. We use different sampling methods: *i.* random, *ii. uniform* and *iii.* timestep selector. As discussed, the output of the timestep selector is a per-timestep binary value $\hat{\alpha} \in \{0, 1\}$ of whether to consider or discard this timestep. So, if T timesteps are processed by the selector, it is able to choose a subset T' timesteps and discard the others, where $T' \ll T$. And to to evaluate the benefit of the selector, the off-the-self classifier then uses only T'.

As shown in figure 4, we observe that the stand-alone selector helps the off-the-shelf classifiers to retain their performance with a reduction of up to 50% of the timesteps. The same improvement is observed for three different CNN classifiers: I3D, ResNet2D and ShuffleNet3D. The main conclusion of this experiment is the following. To realize the efficient recognition of long-range actions, reducing the number of processed timesteps is far more rewarding than reducing the processing



(a) I3D (b) ShuffleNet3D (c) ResNet2D

Figure 4: Our stand-alone Timestep Selector helps improving the performance and reduces the computation of off-the-shelf CNN classifiers – be it 2D/3D heavyweight CNNs or even lightweigh 3D CNNs. More over, if the selector is trained end-to-end with the CNN classifier, the computation is reduced even further.

of each timestep. In other words, our Timestep Selector is able to reduce, by more than half, the computation of the already efficient ShuffleNet3D. See the appendix for full results.

4.3 End-to-end Timestep Selector and Video Classifier

Having demonstrated that a stand-alone selector can benefit off-shelf classifiers, we pose another question – is it possible to train VideoEpitoma end-to-end, given that the selector and the classifier operate on features from two different CNNs, LightNet and HeavyNet, with two different feature spaces. To answer this question, we do the following experiment. We train VideoEpitoma in an end-to-end fashion, where we choose the efficient MobileNet as the LightNet of the selector. As for the HeavyNet of the classifer, we explore multiple choices: I3D, ShuffleNet3D and ResNet2D. Based on our experiments, a careful consideration is to align the timestep features of the 2D LightNet with that of the 3D HeavyNet. In a typical 3D HeavyNet, each timestep is a video snippet of m successive frames $\{f_j, ..., f_{j+m}\}$, represented as one timestep feature $y_i \in \mathbb{R}^{C \times H \times W}$. Thus, the corresponding feature x_i from the 2D LightNet has to be based on the middle frame of the snippet, i.e. frame $f_{j+(m/2)}$.

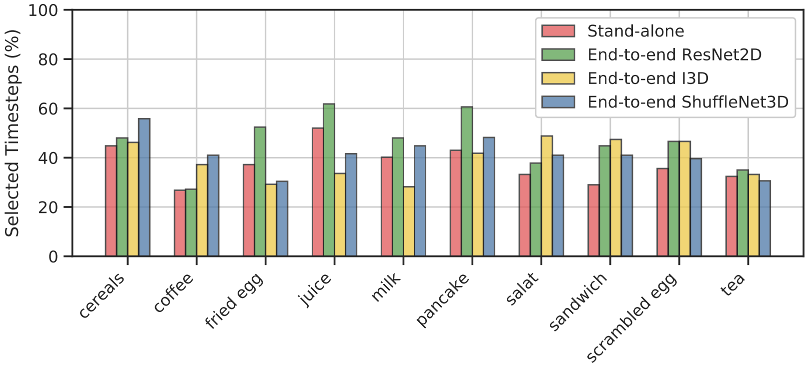


Figure 5: The ratio of selected timesteps for the action categories of Breakfast. When VideoEpitoma is trained end-to-end, the Timestep Selector learns a better selection to the benefit of the classifer. Notice that the selection ratio changes from stand-alone selector (red) to end-to-end training with the HeavyNets: ResNet2D (green) I3D (yellow) and ShuffleNet3D (blue).

The findings of this experiment are as follows. Figure 5 shows the ratio of the selected timesteps by the selector for the videos of each action category of Breakfast. The ratios of the stand-alone (red) is changed when it is trained end-to-end with different HeavyNet: ResNet2D, (blue), I3D (yellow), and ShuffleNet3D (blue). Also, we observe that the choices for the selector when the HeavyNet is 3D CNN tends to agree, relgardless of which 3D CNN is used. Between yellow and blue, we see agreement for 8 of 10 actions. However, the choices tend to vary between 2D and 3D as HeavyNet. Between green and yellow, there is agreement for 3 our of 10 actions. From this experiment, we conclude that, the gating module, depending on LightNet features, learns to select better timesteps to the benefit of the HeavyNet classifier.

4.4 Context Conditional Gating

Gating irrelevant visual evidences is of a great importance in recognizing long-range actions. For example, when discriminating two action categories “Making Pancake” and “Preparing Coffee”, we want to make a gating decision for the visual evidences of ”Knife” and ”Pan”. It is better to discard “Knife” as it is irrelevant to both of actions – this is called frame gating. However, the visual evidence of “Pan” is relevant to only “Making Pancake”. Thus, it’s optimal to consider it only if the action is “Making Pancake” and discarding it otherwise – this is called context gating.

In the Timestep Selector, see figure 2 bottom, we use a temporal modeling layer before the gating module. It enables the correlation between a timestep feature, and the video context, *i.e.* the other timestep features. As a result, the gating mechanism becomes conditioned on both the timestep and the video context. To verify this assumption, we conduct an ablation study. We train a variant of the Timestep Selector without the temporal modeling layer, which makes the gating mechanism conditioned on only the timestep feature.

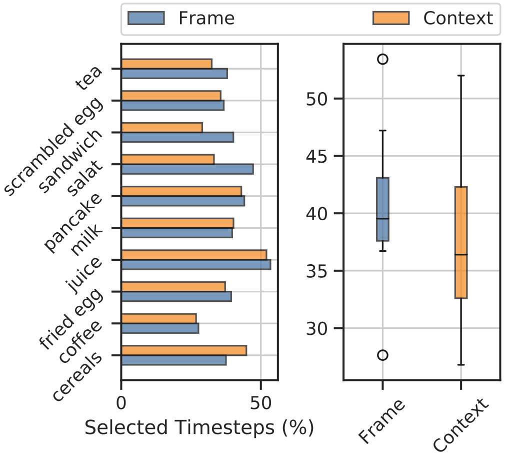


Figure 6: In the Timestep Selector, the gating meachnism is conditioned on both the timestep-level feature and the video-level context, which results is a better conditional gating. If the gating is only frame-conditioned, the ratios of the selected timesteps for action categories have small variance. Which means the gating is less dependent on the context, *i.e.* the action category. On contrary, the notice a big variance for the frame and context-conditioned. The gating becomes more dependent on the action category when selecting the timesteps.

We observe a drop in performance when using this variant of the Timestep Selector. The reason is that when the gating is conditioned only on the timestep feature, it acts as a saliency selector. That is to say, the gating discards only the frames not related to any of the action categories of the dataset. Figure 6, left, shows the ratio of selected timesteps for each action categories of Breakfast. The frame-conditioned gating (blue) tends to select similar ratios regardless of the category. In contrast, we see more diverse ratios for the timestep and context-conditioned gating. Figure 6, right, shows the ratio variances for the two gating mechanisms. The much higher variance for context gating means that it is more dependent on the action category than the frame gating. We conclude that the cost of selecting timestep using LightNet is marginal to that of the HeavyNet and classifier.

4.5 Computation-Performance Tradeoff

When it comes to the recognition of long-range actions, the golden rule is the more timesteps the better the accuracy, and the heavier the computation. But given the huge redundancies of the visual evidences in these timesteps, there is a tradeoff between accuracy and computation. In this experiment, we explore what is effect of this tradeoff on VideoEpitoma, and we compare against off-the-shelf CNNs. Figure 7 shows this tradeoff for three different CNNs: I3D, ResNet2D and ShuffleNet3D. While table 1 details the exact computational budget of VideoEpitoma *v.s.* the competing CNN.

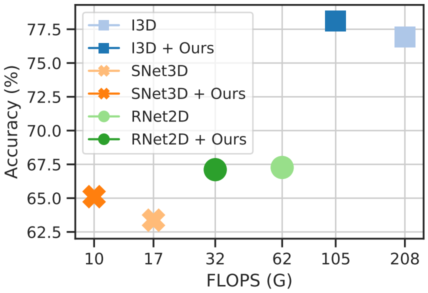


Figure 7: VideoEpitoma, with end-to-end selector, reduces the computation of CNNs by selecting less timesteps.

The conclusion of this experiment is twofold. First, when it comes to classifying the minutes-long actions, classifying a handful of carefully selected timesteps, using VideoEpitoma, is far more rewarding solution than efficiently all of them, using for example ShuffleNet3D. Second, the cost of selecting these timesteps can be significantly reduced by using a lightweight 2D CNN, as MobileNet.

	Timesteps		FLOPS (G) ↓				Accuracy ↑
	LightNet	HeavyNet	LightNet+Gating	HeavyNet	Classifier	Total	
RNet2D	—	16	—	61.7	0.13	61.8	67.27
RNet2D + Ours	16	8	0.94	30.8	0.13	**31.9**	**68.02**
SNet3D	—	16	—	17.2	0.13	17.3	63.37
SNet3D + Ours	16	8	0.94	8.6	0.13	**9.6**	**65.11**
I3D	—	16	—	207.7	0.13	207.7	76.91
I3D + Ours	16	8	0.94	103.8	0.13	**104.8**	**78.11**

Table 1: Breakdown of computation of VideoEpitoma *v.s.* baseline CNNs. We report 3 different types of HeavyNet: *i.* ResNet2D (RNet2D), *ii.* ShuffleNet3D (SNet3D) and *iii.* I3D. The computational cost of LightNet and the gating module is marginal compared to that of the HeavyNet. In addition, our selector retains the performance of the HeavyNet but with using half of the timesteps and almost half of the computational cost.

4.6 Experiments on Charades

Our final experiment is to experiment how VideoEpitoma would fair against off-the-shelf CNN for recognizing the multi-label action videos of Charades. Charades differs from Breakfast in two ways. *i* Videos of Charades are mid-range, with 0.5 minutes as average length, compared to 2 minutes of Breakfast. *ii* Charades is multi-label classifications, with 7 labels per video, and 157 labels in total. Breakfast is single-label classification, with 10 labels in total. Due to these two differences, it is harder to select unrelated timesteps from the videos of Charades than Breakfast – most of the timesteps are already relevant to recognizing the mid-range videos of Charades. Still, VideoEpitoma outperforms the off-the-shelf ResNet2D, at different time scales, see figure 8.

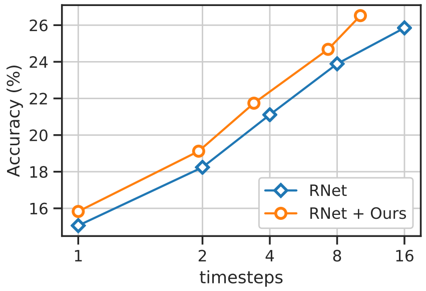


Figure 8: VideoEpitoma improves the performance of the off-the-shelf ResNet2D for recognizing the actions of Charades at different time scales.

5 Conclusion

In this paper, we proposed VideoEpitoma, a neural model for efficient recognition of long-range actions in videos. We stated the fundamental differences between long-range actions and their short-range counterparts (Hussein et al., 2019a;b). And we highlighted how these differences influenced our way of find a solution for an efficient recognition of such videos. The outcome of this paper is VideoEpitoma, a neural model with the ability to retain the performance of off-the-shelf CNN classifiers at a fraction of the computational budget. This paper concludes the following. Rather than building an efficient CNN video classifier, we opted for an efficient selection of the most salient parts of the video, followed by an effective classification of only these salient parts. For a successful selection, we proposed a novel gating module, able to select timesteps conditioned on their importance to their video. We experimented how this selection benefits off-the-shelf CNN classifiers. Futher more, we showed how VideoEpitoma, *i.e.* both the selector and the classifier, improves even further when trained end-to-end. Finally, we experimented VideoEpitoma on two benchmarks for long-range actions. We compared against realted methods to hightight the efficiency of videoEpitoma for saving the computation, and its effectiveness of recognizing the long-range actions.

References

Babak Ehteshami Bejnordi, Tijmen Blankevoort, and Max Welling. Batch-shaped channel gated networks. In *arXiv*, 2019.

Shweta Bhardwaj, Mukundhan Srinivasan, and Mitesh M Khapra. Efficient video classification using fewer frames. In *CVPR*, 2019.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

Zhourong Chen, Yang Li, Samy Bengio, and Si Si. You look twice: Gaternet for dynamic filter selection in cnns. In *CVPR*, 2019.

Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *CVPR*, 2017.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *NeurIPS*, 2015.

Babak Hassibi, David G Stork, and Gregory J Wolff. Optimal brain surgeon and general network pruning. In *ICNN*, 1993.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *arXiv*, 2019.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. In *arXiv*, 2017.

Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Timeception for complex action recognition. In *CVPR*, 2019a.

Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Videograph: Recognizing minutes-long human activities in videos. In *arXiv*, 2019b.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *arXiv*, 2016.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. In *arXiv*, 2017.

Okan Köpüklü, Neslihan Kose, Ahmet Gunduz, and Gerhard Rigoll. Resource efficient 3d convolutional neural networks. In *arXiv*, 2019.

Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. *arXiv*, 2019.

Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *CVPR*, 2014.

Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *arXiv*, 2016.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. In *arXiv*, 2017.

Christos Louizos, Max Welling, and Diederik P Kingma. Learning sparse neural networks through l_0 regularization. In *arXiv*, 2017.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *arXiv*, 2016.

Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. Weakly supervised action localization by sparse temporal pooling network. In *CVPR*, pp. 6752–6761, 2018.

Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. In *arXiv*, 2018.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

Konrad Schindler and Luc Van Gool. Action snippets: How many frames does human action recognition require? In *CVPR*, 2008.

Rakshith Shetty, Marcus Rohrbach, Lisa Anne Hendricks, Mario Fritz, and Bernt Schiele. Speaking the same language: Matching machine to human captions by adversarial training. In *ICCV*, 2017.

Gunnar A Sigurdsson, Gúl Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. In *arXiv*, 2012.

Elzbieta Szelag, Magdalena Kanabus, Iwona Kolodziejczyk, Joanna Kowalska, and Joanna Szuchnik. Individual differences in temporal information processing in humans. In *ANE*, 2004.

Andreas Veit and Serge Belongie. Convolutional networks with adaptive inference graphs. In *ECCV*, 2018.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *arXiv*, 2017.

Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *CVPR*, pp. 3156–3164, 2017.

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 *ECCV*, 2016.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, 2016.

Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, 2018.

Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. In *arXiv*, 2016.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *CVPR*, 2018.

A Appendix

Results on Breakfast

HeavyNet	Sampling	Timesteps							
		1	2	4	8	16	32	64	128
I3D	Uniform	20.83	58.94	67.01	74.05	76.91	84.38	84.11	84.90
	Random	42.75	53.13	60.96	69.22	75.60	81.45	83.52	83.81
	Selector	48.34	60.74	68.74	79.69	82.81	85.68	86.46	87.50
ShuffleNet3D	Uniform	18.75	38.54	46.27	60.76	63.37	69.62	67.27	66.23
	Random	36.34	45.81	50.50	58.00	60.90	67.17	66.57	65.82
	Selector	39.93	50.23	54.09	63.11	64.41	69.36	68.32	66.49
ResNet2D	Uniform	27.00	50.69	61.02	67.01	67.27	72.22	74.57	76.13
	Random	41.37	51.80	57.43	62.97	63.74	70.97	72.95	74.28
	Selector	46.63	56.28	63.89	68.02	69.62	73.15	74.38	74.98

Table 2: Our stand-alone timestep selector helps improving the performance of off-the-shelf CNN video classifiers, regardless of the CNN used – be it 2D CNN as ResNet, heavyweight 3D CNN as I3D or even lightweight 2D CNN as ShuffleNet3D.