AutoShot: A Short Video Dataset and State-of-the-Art Shot Boundary Detection

Wentao Zhu[1], Yufang Huang[3], Xiufeng Xie[1], Wenxian Liu[1],
Jincan Deng[1], Debing Zhang[1], Zhangyang Wang[2], Ji Liu[1]
[1]Kuaishou Technology [2]University of Texas at Austin [3]Cornell University

Abstract

The short-form videos have explosive popularity and have dominated the new social media trends. Prevailing short-video platforms, e.g., Kuaishou (Kwai), TikTok, Instagram Reels, and YouTube Shorts, have changed the way we consume and create content. For video content creation and understanding, the shot boundary detection (SBD) is one of the most essential components in various scenarios. In this work, we release a new public Short video sHot bOundary deTection dataset, named SHOT, consisting of 853 complete short videos and 11,606 shot annotations, with 2,716 high quality shot boundary annotations in 200 test videos. Leveraging this new data wealth, we propose to optimize the model design for video SBD, by conducting neural architecture search in a search space encapsulating various advanced 3D ConvNets and Transformers. Our proposed approach, named AutoShot, achieves higher F1 scores than previous state-of-the-art approaches, e.g., outperforming TransNetV2 by 4.2%, when being derived and evaluated on our newly constructed SHOT dataset. Moreover, to validate the generalizability of the AutoShot architecture, we directly evaluate it on another three public datasets: ClipShots, BBC and RAI, and the F1 scores of AutoShot outperform previous state-of-the-art approaches by 1.1%, 0.9% and 1.2%, respectively. The SHOT dataset and code can be found in https://github.com/wentaozhu/AutoShot.git.

1. Introduction

Short-form videos have been widely digested among the entire age groups all over the world. The percentage of short videos and video-form ads has an explosive growth in the era of 5G, due to the richer contents, better delivery and more persuasive effects of short videos than the image and text modalities [36]. This strong trend leads to a significant and urgent demand for a temporally accurate and comprehensive video analysis in addition to a simple video classification category [39,42]. Shot boundary detection is a fundamental component for temporally comprehensive video analysis and can be a basic block for various tasks, *e.g.*, scene boundary detection [6,26], video structuring [36], and event segmentation [29]. For instance, rewarded videos can be automated created of desired lengths for different platforms, leveraging the accurate shot boundary detection in the intelligent video creation.

To accelerate the development of video temporal boundary detection, several datasets have been collected with laboriously manual annotation. Conventional shot boundary detection datasets, *e.g.*, BBC Planet Earth Documentary series [1] and RAI [2], only consist of documentaries or talk shows where the scenes are relatively static. Tang *et al*. [32] further contribute a large-scale video shot database, ClipShots, consisting of different types of videos collected from YouTube and Weibo covering more than 20 categories, including sports, TV shows, animals, *etc*. Shou *et al*. [29] construct a generic event boundary detection (GEBD) dataset, Kinetics-GEBD, which defines a clip as the moment where humans naturally perceive an event. Since the video lengths of short and conventional videos differ extensively, *i.e.*, 90% short videos of length less than one minute *versus* videos in other datasets having length of 2-60 minutes as shown in Table 1 and Fig. 1 Right, it dramatically leads to significant content, display, temporal dynamics and shot transition differences as shown in Fig. 1 Left. A short video dataset is necessary to accelerate the development and proper evaluation of short video based shot boundary detection.

On the other hand, several endeavors have been made to improve the accuracy of video shot boundary detection (SBD). DeepSBD [11] firstly applies a deep spatio-temporal ConvNet to the video SBD. Deep structured model (DSM) [32] designs a cascade framework to accelerate the speed of SBD. TransNet [21] uses dilated convolutional cells [38] to process a sequence of resized frames. TransNetV2 [30] incorporates techniques, *e.g.*, convolution kernel factorization [37], batch normalization [14], skip connection [12], and further improves F1 scores on ClipShots [32] and BBC [1].

In this work, we firstly collect a short video dataset, named SHOT, consisting of 853 short videos with 11,606

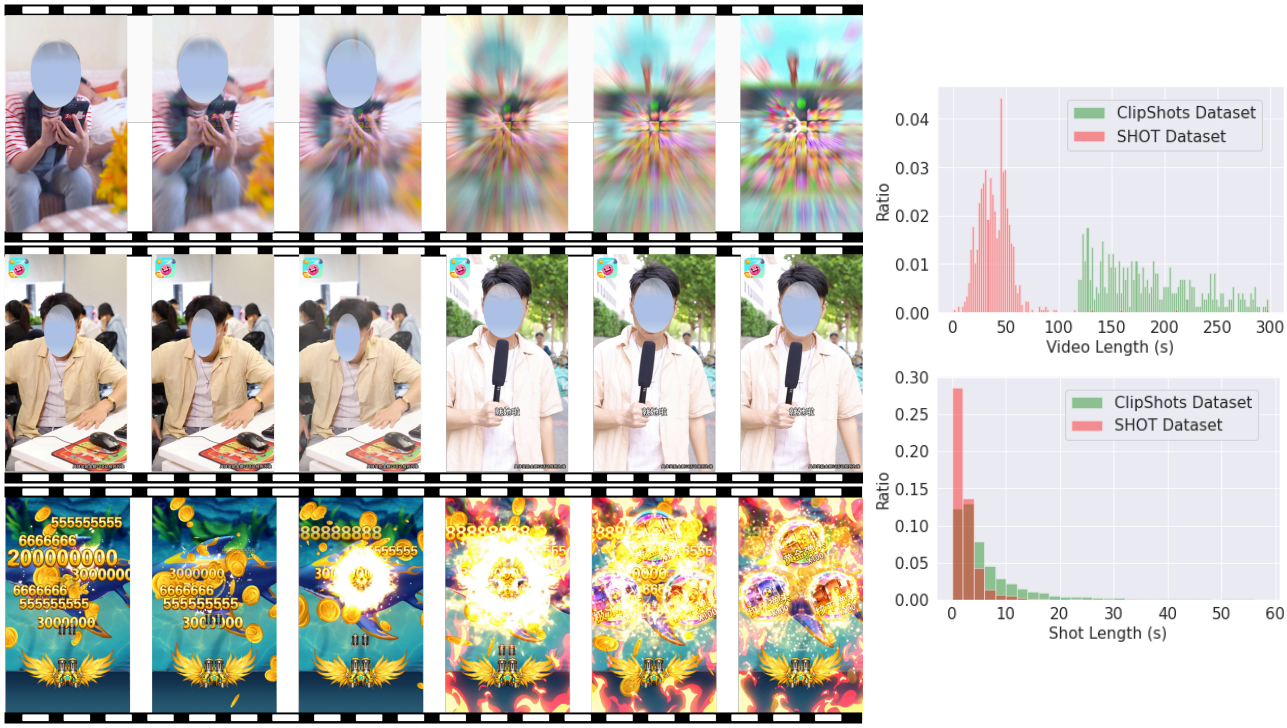


Figure 1. Left: Detecting a shot boundary can be a challenging task in short videos. The shot transition can be a combination of several complicated gradual transitions (the first row) and a quick transition of the subject in two shots (the second row). The visual effect of the intra-shot can vary greatly in game videos (the third row). Right: Video and shot length (s) comparison of test sets in ClipShots and our collected SHOT. There rarely has a video length range overlap between short videos in SHOT and test videos in ClipShots (up). The shot lengths of short videos are within six seconds, while the shot lengths of ClipShots can range from two seconds to 30 seconds (bottom).

manually shot boundary fine annotations. The 200 test videos with 2,716 shot boundary annotations are labeled by experts with two rounds. Leveraging this new data wealth, we aim to improve the accuracy of video shot boundary detection, by conducting neural architecture search [41] in a search space encapsulating various advanced 3D ConvNets [25, 40, 43, 45] and Transformers [35, 39]. Single path one-shot SuperNet strategy [9] and Bayesian optimization [28] are employed. The searched model, named AutoShot, outperforms TransNetV2 by 4.2% on our SHOT in terms of F1 score, and by 3.5% in terms of precision metric with a fixed recall rate as TransNetV2, respectively. We further evaluate the searched AutoShot architecture on ClipShots, BBC and RAI, and F1 score of AutoShot surpasses previous state-of-the-art approaches by 1.1%, 0.9% and 1.2%, respectively. Our contributions are summarized as follows:

- We collect a short video shot boundary detection dataset (SHOT), which consists of 853 short videos and 11,606 shot boundary annotations. The SHOT will be released and can be employed to advance the development of various short video understanding tasks.
- We design a video shot boundary detection search space encapsulating various advanced 3D ConvNets and Transformers, and build a neural architecture search pipeline for shot boundary detection.
- The searched model, named AutoShot, proves to be a highly competitive shot boundary detection architecture, that significantly outperforms previous state-of-the-art approaches not only on its derived SHOT dataset, but also on other public benchmarks.

To the best of our knowledge, the collected SHOT dataset is the first dataset for short video shot boundary detection, and AutoShot is the firstly specially designed neural architecture search method for shot boundary detection.

2. Related Work

Extensive efforts have been made to collect video *shot* boundary detection datasets [1, 2, 4, 16, 27, 32], which have significantly accelerated the development of advanced *shot* boundary detection methods. A specific type of video boundary detection attempts to parse the video into pieces of human actions, where instructional videos with human performing diverse actions are commonly used [18, 31, 33]. Another widely studied type of video boundary detection

is *scene* boundary detection, where videos are split into several semantically independent clips. Movies and TV episodes are commonly used for *scene* boundary detection, and MovieScenes [26] and AdCuepoints [6] are two large-scale movie datasets for *scene* boundary detection. Furthermore, Shou *et al*. [29] collect a new benchmark, Kinectics-GEBD, for generic *event* boundary detection. For video ads *scene* boundary detection, Wang *et al*. [36] recently collect a multi-modal video ads understanding dataset, which involves *scene* boundary detection and multi-modal scene classification. However, *scene* boundary detection is typically based on the pre-extracted *shots* and the evaluation of scene segmentation is *shot*-level instead of frame-level.

The TV series and talk shows have also been used for *shot* boundary detection, where BBC Planet Earth Documentary series [1] and RAI [2] are two commonly used datasets consists of tens of videos having length from half an hour to one hour. ClipShots [32] enhances the conventional *shot* boundary detection datasets by collecting diverse videos from various media platforms, *e.g.*, YouTube and Weibo, and it is one of the most challenging large-scale *shot* boundary detection datasets. The shot transitions in short videos, are quite different from that of movies as shown in Fig. 1 Right and Table 1, and it is extremely necessary that a short video dataset is collected to advance the development of short video *shot* boundary detection. The only short video dataset publicly available so far, to our best knowledge, is the SVD dataset [15]; yet that is developed for a completely different task for near-duplicate video retrieval.

The accuracy of *shot* boundary detection has been improved, leveraging the aforementioned high quality datasets and deep learning. Before the deep learning era, PySceneDetect [3] is a popular *shot* boundary detection library, which relies on conventional features, *e.g.*, changes between frames in the HSV color space. Recent progresses on video boundary detection can be divided into two categories, *scene* boundary detection and *shot* boundary detection. Rao *et al*. [26] propose a local-to-global *scene* segmentation framework integrating multi-modal information across clip, segment, and movie. Chen *et al*. [6] further propose a shot contrastive self-supervised learning [44] to learn a shot representation that maximizes the similarity between nearby shots compared to randomly selected shots, then apply the learned shot representation for *scene* boundary detection.

The *scene* boundary detection highly depends on the accurate *shot* boundary detection. DeepSBD [11] predicts a likelihood of transitions in a clip of 16 frames by the C3D network [34]. DSM [32] utilizes a cascade framework to accelerate the speed of *shot* boundary detection. Gygli [10] constructs a fast *shot* boundary detection without any post-processing. TransNet [21] uses dilated convolution blocks [38] and achieves comparable accuracy as DeepSBD without post-processing. TransNetV2 [30] surpasses previous state-of-the-art approaches with advanced components, *e.g.*, skip connection [12], batch normalization [14], kernel factorization [37], frame similarities as features and multiple classification heads. Leveraging the progress of 3D ConvNets [25], Transformers [35] and neural architecture search [9], AutoShot automatically identifies the optimal *shot* boundary detection architecture from the designed search space, which achieves better accuracy than previous methods on the collected SHOT, ClipShots, BBC and RAI datasets.

3. SHOT: Short Video Shot Boundary Detection Dataset

Short video is one of the most prevailing medias in these days because of its richer contents and more vivid effects than its conventional counterparts, *i.e.*, pure text and static picture medias. The easy and affordable access to fast mobile networks in the 5G era accelerates the widespread adoption of short video platforms, *e.g.*, Instagram Reels, YouTube Shorts, and TikTok. With the large number of users and video ads in these main-stream short video platforms, it is critical to advance the current development of video temporal segmentation, especially short video shot boundary detection task, which is a fundamental task for many following semantic understanding tasks.

3.1. Challenges of Short Video Shot Boundary Detection

A short video is typically defined as a video of length less than two minutes. The short video length leads to a much easier spread of these videos and more popular short videos than conventional movies of hours long. On the other hand, the short video length forces the whole events to occur in a short time period, which causes much faster pace of events. This in turn leads to a much shorter length of a shot in the short video as shown in Fig. 1 Right, which aggregates the difficulty of short video shot boundary detection.

The giant difference of video lengths between the test sets in the collected SHOT dataset and ClipShots [32] is visualized in Fig. 1 Right. Almost all the test short videos have lengths less than 100 seconds, while almost all the test videos in the ClipShots have lengths greater than 120 seconds. The short total video length directly leads to rapid shot transitions in the SHOT dataset as indicated in Fig. 1 Right. Most shot lengths are within five seconds in the test set of SHOT dataset, and the shot lengths of test set in the ClipShots can range from two seconds to 30 seconds. The conventional shot boundary detection dataset may be inappropriate for the development of short video shot boundary detection because of the great difference of video and shot length distributions.

Short video shot boundary detection is much more challenging and difficult as illustrated in Fig. 1 Left and Fig. 2 because the scene of the short video is much more complicated than conventional videos. For instance, the shot transition commonly utilizes a combination of several complicated shot gradual transitions for the persuasive effect in the short video (the first row). The second common challenge is for vertically ternary structured videos (the second row of Fig. 2), where only the middle part of the video changes. The uppermost and lowermost regions display the download link or brand in the video ads. The vertically ternary structured video increases shot boundary detection difficulty greatly due to the relatively small region change in the squeezed content region. The exaggerated expression in the virtual scene causes false alarms in the game video shot boundary detection (the third row). Actually, the game video takes a large proportion of video ads and short videos. Therefore, collecting a short video dataset is nontrivial for the challenging short video shot boundary detection.

3.2. SHOT Dataset

To accelerate the study of short video related shot boundary detection, we collect 853 short videos from one of the most widely used short video platforms. The dataset property comparison is listed in Table 1. The total number of frames is 960,794, close to one million frames. The frame-wise shot boundary annotation is a heavy task. The data annotation and quality control strategy are in appendix. To remove human private information, we employ the state-of-the-art face detector [7,24] to detect and obscure the human face region.

Inspired by the frame processing in TransNetV2 [30], we develop a video thumbnail image based annotation by resizing each frame to be 48×27 as shown in Fig. 3. The frame number is adaptively displayed in the upper left corner of each frame, which significantly reduces the annotator's efforts for frame number check. If the pixel value is dark in the frame number position, we display the frame number in light color. Otherwise, we display the frame number in dark color. We have three experts and an annotation team to complete all the annotation of 960,794 frames. The annotation team conducts the annotation for 459 short videos and obtains 6,111 shots totally. After the annotation, we conduct an expert inspection for the 6,111 annotations. We randomly choose 200 shot annotations, and find that 2% error rate for the 6,111 shot annotations exists. Considering the ambiguity of the shot definition and the annotation difference of annotators, we believe that the 2% error rate of the 6,111 shots is acceptable.

The quality of annotations on the test set directly affects the accurate evaluation of benchmark methods, *i.e.*, the quality of the short video dataset. To guarantee the high quality of the shot boundary annotations on the test set, we randomly choose 200 short videos from the rest 394 short videos annotated by the three experts as the test set.

Dataset	BBC	RAI	Clip.	SHOT
Complex Grad. Trans.	✗	✗	✗	✓
Virt. Scene	✗	✗	✗	✓
Tern. Video	✗	✗	✗	✓
Avg. Video Len. (s)	2945	591	237	39.5
Avg. Shot Len. (s)	6.57	5.65	15.34	2.59

Table 1. Comparison of different short boundary detection datasets, *i.e.*, BBC, RAI, ClipShots and SHOT, *w.r.t.* complex gradual transition, virtual scene, ternary video, average video length and average shot length.

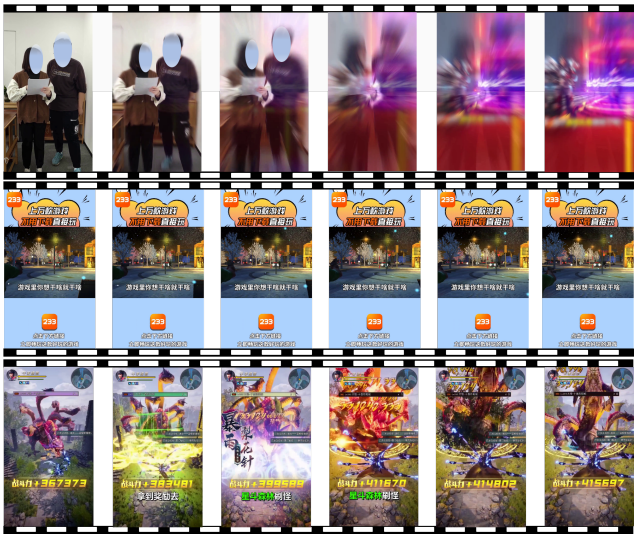


Figure 2. Unique challenges of short video shot boundary detection in SHOT, *e.g.*, a combination of complicated shot gradual transitions, vertically ternary structured video, and great intra-shot change in virtual scenes of game video.

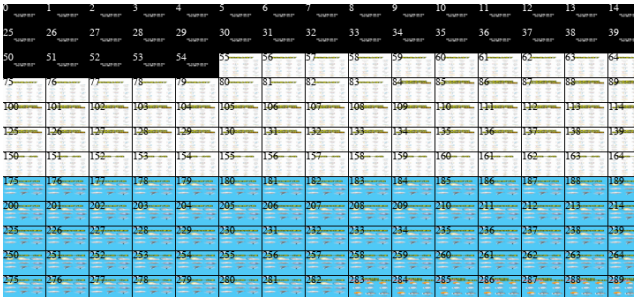


Figure 3. A thumbnail image for one case from our collected SHOT dataset. The frame number is displayed in the upper left corner of each frame.

For the 200 test short videos, we employ two round annotations, where the first round yields 2,616 shots. In the second round, we conduct a rigorous check for the annotations. In addition to fixing a handful of false positive annotations, which cannot be corrected by our manually designed rules for the automated inspection, the second round totally yields 2,716 shots, *i.e.*, re-collecting 100 shots from false negatives.

The cases from the SHOT dataset can be found in Fig. 1 Left, Fig. 2, and Fig. 5. The annotation records the start and end frame numbers of each shot, which is visualized by the pink and light/white colors in Fig. 5. All the training and testing short videos, shot boundary annotations, the evaluation metric scripts and the explicit video-level data split will be publicly available.

4. Automated Shot Boundary Detection

4.1. Shot Boundary Detection Search Space Design

We design a SuperNet based on one of the previous state-of-the-art approaches, TransNetV2 [30], where the feature representational learning network can be considered as a sequence of six factorized dilated deep 3D convolutional neural network (DDCNNV2) blocks. Leveraging the advanced 3D ConvNets [25] and Transformers [35], we deign a shot boundary detection neural architecture search space. Specifically, AutoShot conducts architecture search on seven blocks, where we add a self-attention layer number search after the sixth block. The search blocks in the first six blocks are illustrated in Fig. 4. We firstly design four types of factorized 3D convolutions in the search space. Let $\mathbf{x}$ be the input for the current block, the four kinds of search blocks can be formulated as following:

(1) DDCNNV2: We conduct the search on the number of dilated convolution branches n_d, *i.e.*, 4 and 5, and the channel number of 2D spatial convolution n_c, *i.e.*, 1, 2 and 3 times of the input channel numbers, for the original DDCNNV2 block in the TransNetV2 as illustrated in Fig. 4a. The DDCNNV2 can be formulated as

$$\begin{aligned} \mathbf{h} &= \text{ReLU}(\text{BN}(\text{Concat}([\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_{n_d}]))), \\ \mathbf{h}_i &= (\mathbf{T}_i \cdot \mathbf{S}_i) \cdot \mathbf{x} = \mathbf{T}_i(\mathbf{S}_i(\mathbf{x})),\ i = 1, \cdots, n_d, \end{aligned} \tag{1}$$

where $\mathbf{h}$ is the output of the current block, $\mathbf{S}_i$ is the 2D spatial convolution with the channel number $\lceil \frac{n_c}{n_d} \rceil$, $\mathbf{T}_i$ is the 1D temporal convolution with the channel number $\lceil \frac{4F}{n_d} \rceil$ and dilation rate 2^{i-1}, and F is a pre-defined channel number in Fig. 4a. The key components in DDCNNV2 are dilated temporal 1D convolutions, and factorized 3D convolutions with spatial 2D convolutions and temporal 1D convolutions. The design enforces diverse contextual temporal feature extraction and reduces the number of learnable parameters, which might reduce the over-fitting.

(2) DDCNNV2A: To unify the feature extractor of the spatial 2D convolutions, we can employ a shared 2D convolution instead of multiple branches of spatial 2D convolutions, as illustrated in Fig. 4b. The shared spatial 2D convolution aims to extract a unified spatial feature for the following diverse temporal feature extractions. The DDCNNV2A can be expressed as

$$\begin{aligned} \mathbf{h} &= \text{ReLU}(\text{BN}(\text{Concat}([\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_{n_d}]))), \\ \mathbf{h}_i &= (\mathbf{T}_i \cdot \mathbf{S}) \cdot \mathbf{x} = \mathbf{T}_i(\mathbf{S}(\mathbf{x})),\ i = 1, \cdots, n_d, \end{aligned} \tag{2}$$

where $\mathbf{S}$ is a shared 2D spatial convolution with searched channel number n_c.

(3) DDCNNV2B: Inspired by the design of Pseudo-3D network [25], we construct another two search blocks to learn various spatio-temporal representations as illustrated in Fig. 4c and 4d. The DDCNNV2B can be given by

$$\begin{aligned} &\mathbf{h} = \text{ReLU}(\text{BN}((\mathbf{S} + \mathbf{T}) \cdot \mathbf{x})) = \text{ReLU}(\text{BN}(\mathbf{S}(\mathbf{x}) + \mathbf{T}(\mathbf{x}))), \\ &\mathbf{T}(\mathbf{x}) = \text{Concat}([\mathbf{T}_1(\mathbf{x}), \mathbf{T}_2(\mathbf{x}), \cdots, \mathbf{T}_{n_d}(\mathbf{x})]). \end{aligned} \tag{3}$$

To ensure the channel numbers of spatial features and temporal features are equal, the channel number of 2D spatial convolution is fixed to be four times of the current block's input dimension number $4F$.

(4) DDCNNV2C: Different from DDCNNV2B, the temporal convolution of DDCNNV2C can still utilize the feature of spatial convolution as illustrated in Fig. 4d, which can be formulated as

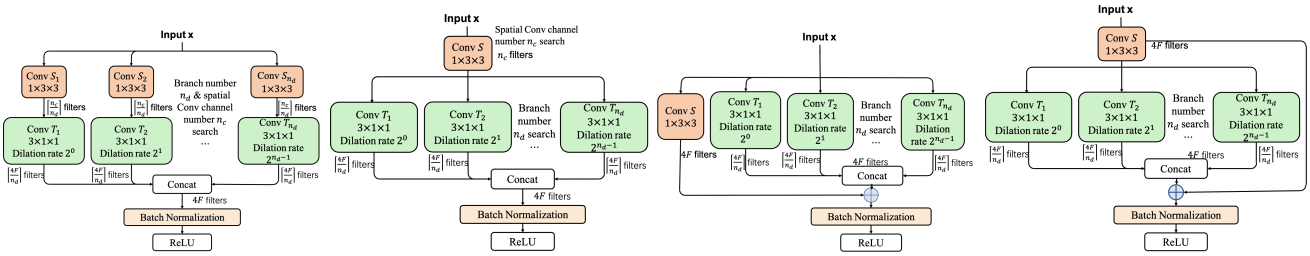
$$\begin{aligned} \mathbf{h} &= \text{ReLU}(\text{BN}((\mathbf{S} + \mathbf{T} \cdot \mathbf{S}) \cdot \mathbf{x})) \\ &= \text{ReLU}(\text{BN}(\mathbf{S}(\mathbf{x}) + \mathbf{T}(\mathbf{S}(\mathbf{x})))), \\ \mathbf{T}(\mathbf{S}(\mathbf{x})) &= \text{Concat}([\mathbf{T}_1(\mathbf{S}(\mathbf{x})), \mathbf{T}_2(\mathbf{S}(\mathbf{x})), \cdots, \mathbf{T}_{n_d}(\mathbf{S}(\mathbf{x}))]). \end{aligned} \tag{4}$$

We further construct a 1D temporal Transformer block after six factorized convolution layers to enhance the temporal modeling, whose input is a flattened frame-wise convolutional feature. We conduct the number of self-attention layers search $\{0, 1, 2, 3, 4\}$ in the Transformer block.

In summary, AutoShot has seven search blocks. In the first six search blocks, it conducts the channel number search and branch/dilation number search for DDCNNV2 and DDCNNV2A. Limited by the dimension number consistency of element-wise addition, DDCNNV2B and DDCNNV2C searches the branch number. Consequently, we have $3 \times 2 \times 2 + 2 \times 2 = 16$ options for each search block in the first six search blocks. The search space of AutoShot totally has $(16^6) \times 5 = 8.39 \times 10^7$ candidate architectures.

4.2. AutoShot Training and Search

After the construction of the base network for representational learning, we concatenate representations from the base network with RGB histogram similarity of raw input frame and learnable cosine similarity of concatenated block



(a) DDCNNV2 (b) DDCNNV2A (c) DDCNNV2B (d) DDCNNV2C

Figure 4. Illustration of the search blocks for the first six blocks in the AutoShot. It consists of four types of factorized 3D convolution blocks, (a) DDCNNV2 with 2D spatial convolutions followed by 1D temporal convolutions, (b) DDCNNV2A with a shared 2D spatial convolution and 1D temporal convolutions, (c) DDCNNV2B accumulating a 2D spatial convolution and 1D temporal convolutions, and (d) DDCNNV2C compromising between DDCNNV2A and DDCNNV2B.

features [5, 30] to construct a 4,864 dimension feature vector. Then a fully connected layer of 1,024 neurons with ReLU activation [23] is added. Next a dropout layer [17] with a dropout rate 0.5 is used before the final two frame-wise classification heads for a single middle frame of a transition y and all transition z.

The SuperNet training of AutoShot utilizes an efficient weight sharing strategy. The weight sharing strategy [9, 20] encodes the search space in a SuperNet, and all the candidate architectures share the weights of the SuperNet. One shot NAS [9] decouples the SuperNet training and architecture search, which yields a better accuracy. We also conduct two sequential steps for SuperNet training and architecture search. We utilize a single path and uniform sampling strategy to reduce the co-adaptation between node weights [9].

We implement a Bayesian optimization [28] based architecture search for AutoShot. Bayesian optimization iterates between fitting probabilistic surrogate models and determining which configuration to evaluate next by maximizing an acquisition function. Gaussian process with a Hamming kernel is utilized as the surrogate function. We employ a random exploration in the initialization to obtain a good Gaussian process model. For the acquisition function, we use probability of feasibility [8].

$$\boldsymbol{p}|\boldsymbol{a} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{K}), \quad \text{Acc}|\boldsymbol{p}, \sigma^2 \sim \mathcal{N}(\boldsymbol{p}, \sigma^2 \boldsymbol{I}), \tag{5}$$

where variables $\boldsymbol{p}$ are jointly Gaussian, $\boldsymbol{a}$ are a set of observed architectures, Hamming kernel $K_{ij} = k(a_i, a_j)$, and Acc are the evaluated accuracy metric, *i.e.*, F1 or precision with a fixed recall, for these architectures with weight sharing. The parameters of GP, $\boldsymbol{\mu}$ and σ, can be estimated by maximizing the marginal log-likelihood.

For the candidate architecture retraining, we firstly employ the same two classification, *i.e.*, single middle frame y of a transition and all transition z, cross-entropy losses as the SuperNet training

$$\mathcal{L}_{retrain} = -\sum_{i=1}^{N}\sum_{j=1}^{N_F} \left[\lambda_1 y_{i,j} \log \hat{y}_{i,j} + \lambda_2 z_{i,j} \log \hat{z}_{i,j}\right], \tag{6}$$

where N is the batch size in the stochastic gradient descent (SGD), N_F is the pre-defined number of frames for each training sample which is processed in each batch of training, λ_1 and λ_2 are trade-offs between two classification heads, and $\hat{y}_{i,j}$ and $\hat{z}_{i,j}$ are two frame-wise predictions of single middle frame of a transition and all transition, respectively.

After retraining with the plain multi-head cross-entropy classification loss in Eq. (6), we further enhance the candidate networks by employing the best performing candidate network as a teacher network in knowledge distillation [13], and utilize weight grafting [22] to further improve the best accuracy. The knowledge distillation is used to align candidate networks with a desired accuracy, and the weight grafting adaptively balances the grafted information among aligned networks, which improves the representation capability and boosts the accuracy. The entropy-based weight grafting can be formulated as

$$\begin{aligned}
&\mathcal{L}(W) = -\sum_{i=1}^{N}\sum_{j=1}^{N_F} \left[\lambda_1 \tilde{y}_{i,j} \log \hat{y}_{i,j} + \lambda_2 \tilde{z}_{i,j} \log \hat{z}_{i,j}\right], \\
&\alpha = A \times (\arctan(c \times (H(W_l^{M_2}) - H(W_l^{M_1})))) + 0.5, \\
&W_l^{M_2} = \alpha W_l^{M_2} + (1-\alpha) W_l^{M_1},
\end{aligned} \tag{7}$$

where $\tilde{y}_{i,j}$ and $\tilde{z}_{i,j}$ are two frame-wise predictions of single middle frame of a transition and all transition from the teacher network, A and c are fixed hyperparameters, α is the coefficient based on entropy to balance networks, $H(\cdot)$ is the entropy based on the bins of network weights $W_l^{M_1}$ or $W_l^{M_2}$, l is the layer in the network, M_1 and M_2 are two networks where network M_2 accepts the information from network M_1.

5. Experiments

We conduct neural architecture search based on the evaluation metrics on the collected SHOT and validate the effectiveness of the searched optimal architecture on SHOT, ClipShots [32], BBC [1], and RAI [2]. In both SuperNet training and candidate network retraining, we construct each training sample by concatenating two shots randomly and the number of frames N_F in each training sample is set as 60. The hyperparameters λ_1, λ_2, A, c, the number of bins in the entropy calculation and the number of grafting networks in Eq. (7) are set as 5, 0.1, 0.4, 1.0, 10 and 3, respectively, which are generally followed the setting of previous works [22, 30]. We use stochastic gradient descent with learning rate 0.1, momentum 0.9, batch size 16 and the number of epochs 12. In the search, the number of populations per epoch is 48 and the total number of epochs is 100 with 20 epochs for the initialization. We utilize the OpenBox library [19] to implement the Bayesian optimization in the search. We use one 32 GB NVIDIA Tesla V100 GPU for both SuperNet and candidate network training, and use eight 32 GB NVIDIA Tesla V100 GPUs to accelerate the search speed. All the code, data and trained models will be released for the reproducibility.

Neural architecture search on SHOT We train the SuperNet on the combined training set of SHOT and ClipShots, and conduct the search based on the evaluation metrics on the collected SHOT dataset. For a fair comparison, we closely follow the dataset protocol and metric calculation of previous work [2, 30]. Additionally, we also conduct a search based on the precision metric given a fixed recall rate 0.71 same as TransNetV2, because a higher F1 metric cannot guarantee higher precision and recall scores simultaneously in practice. From Table 2 Left, AutoShot outperforms TransNetV2 by 4.2% and 3.5% based on F1 and precision metrics. Note that the F1 score of PySceneDetect [3] on SHOT is less than 0.6, which is far behind AutoShot, and it can hardly handle challenging gradual transitions. For AutoShot, we find that 54% of missed shots are gradual transitions, and gradual transitions take 30% of shots in SHOT. Gradual transition has no huge inter-frame difference, which is difficult to be fully detected. In the following, we only compare AutoShot based on the F1 metric with other methods for consistency.

To compare the predictions visually, we employ video thumbnail images to clearly demonstrate the difference. The ground truth boundary is shown in the pink or light/white color as shown in Fig. 5. The detected boundary of AutoShot is marked as pink, cyan or light, and the detected boundary of TransNetV2 is visualized as cyan or light. AutoShot successfully detects minor shot transitions as shown in the pink color, where the TransNetV2 fails. For the clear transitions, both the TransNetV2 and AutoShot succeed, as shown in the light color. The false positives of both TransNetV2 and AutoShot are shown in cyan, which are hard negative shots. Reducing the false positives might require adaptively understanding of contextual semantics across the whole video, and some false positives are ambiguous even for human annotators.

Figure 5. Visual comparison of shot boundaries of ground truth (pink, light/white), TransNetV2 (cyan, light/white), and AutoShot (pink, cyan, light/white) on four clips from SHOT dataset. AutoShot detects minor transition, shown in the pink color. The light/white color denotes that both the TransNetV2 and AutoShot successfully detects the shot boundary. The cyan denotes the false positives of both TransNetV2 and AutoShot, which might require adaptively understanding of contextual semantics across the video.

Generalization on other datasets After obtaining the optimal network architecture from the SHOT dataset, we validate the network generalizability on other three publicly and widely used datasets, ClipShots [32], BBC [1], and RAI [2]. The three existing datasets are quite different from our SHOT dataset, as shown in the section 3. BBC Planet Earth Documentary series [1] consists of 11 episodes from the BBC educational TV series Planet Earth. Each episode is approximately 50 minutes long, and the whole dataset contains around 4900 shots and 670 scenes. RAI [2] dataset is based on a collection of ten randomly selected broadcasting videos from the Rai Scuola video archive, where the length of each video is around half an hour. The ClipShots collects thousands of online conventional videos, not short videos, from YouTube, which is a much more challenging dataset than BBC and RAI. We use the same dataset split and protocol as previous work [2, 30].

From Table 2 Right, simply applying the searched AutoShot architecture to the three datasets obtains better F1 scores than previous state-of-the-art approaches consistently, which sufficiently validates the effectiveness and

Method	TransNetV2	AutoShot @F1	AutoShot @Precision
F1	0.799	**0.841**	0.826
Prec.	0.904	0.923	**0.939**

Dataset	ClipShots	BBC	RAI
DSMs	0.761	0.893	0.928
ST ConvNets	0.759	0.926	0.939
TransNet	0.735	0.929	0.943
TransNetV2	0.776	0.962	0.939
AutoShot	**0.787**	**0.971**	**0.955**

Table 2. Left: AutoShot surpasses TransNetV2 by 4.2% and 3.5% based on the F1 score and precision metric with a fixed recall as TransNetV2. Right: the searched optimal architecture is validated on three widely used shot boundary detection datasets. AutoShot consistently achieves the best F1 compared to DSMs [32], ST ConvNets [11], TransNet [21] and TransNetV2 [30]. The best F1 scores are indicated in **bold**.

AutoShot-	S	M	L
w/o retrain	0.816	0.822	0.831
w/ retrain	0.833	0.837	**0.841**

Method	w/o KD	w/ KD	w/ KD+ weight graft
F1	0.825-0.837	0.832-0.838	**0.841**

Table 3. F1 scores of different search spaces (Left), knowledge distillation (KD) and weight grafting (Right) on the collected SHOT dataset. The best F1 scores are indicated in **bold**.

good generalizability of AutoShot. Specially, AutoShot outperforms previous state-of-the-art approaches by 1.1%, 0.9% and 1.2% on ClipShots, BBC and RAI. Note that we reproduce TransNetV2 based on PyTorch and obtain a F1 score of 0.776 on ClipShots, while the original paper [30] reports 0.779 based on TensorFlow.

Effect of search space We investigate the effect of three different search spaces, *i.e.*, AutoShot-S, -M and -L, in Table 3 (Left). AutoShot-S only employs DDCNNV2A components in the search space, which has six search options per block. AutoShot-M employs DDCNNV2 and DDCNNV2A components in the search space, which has 12 search options per block. The AutoShot-L denotes the search space defined in section 4.1, which achieves the best F1 score. The combined various 3D ConvNet variants, *i.e.*, DDCNNV2, DDCNNV2A, B and C, in each search block improves F1 score in both retraining and candidate architectures after search, since the optimal blocks for different layers vary and more search options in AutoShot permit to identify the optimal composition.

Searched architectures Specifically, the obtained architecture based on F1 score, AutoShot@F1, is DDCNNV2$\{(n_c$=$4F$, n_d=4), A(n_c=$4F$, n_d=5), A(n_c=$4F$, n_d=5), A(n_c=$4F$, n_d=5), (n_c=$12F$, n_d=5), (n_c=$8F$, n_d=5)$\}$, which has floating-point operations (FLOPs) of 37 GMACs. The obtained architecture based on precision metric, AutoShot@Prec., is DDCNNV2$\{(n_c$=$12F$, n_d=4), (n_c=$8F$, n_d=4), B(n_d=4), C(n_d=4), B(n_d=5), B(n_d=4)$\}$, which has FLOPs of 30 GMACs. We find that the optimal architectures indeed employ diverse blocks and less number of operations than TransNetV2 of 41 GMACs. The search on SHOT chooses more dilated convolutional branches n_d and diverse blocks, because more dilated convolutional branches n_d and diverse blocks enhances the representational learning power for various temporal granular shot transitions, which vastly exist in short videos. This is probably another reason that AutoShot on SHOT achieves much bigger improvement than that on the conventional video datasets. Although the two optimal architectures use no self-attention layer, the training of SuperNet and architectures with close F1 or precision scores utilize the self-attention.

Effect of knowledge distillation and weight grafting We ablate knowledge distillation and weight grafting in Table 3 (Right) based on the constructed search space in section 4.1. Without knowledge distillation in Eq. (6), the range of F1 scores after retraining can be 0.825-0.837. We use the best performing, *i.e.*, F1 score of 0.837, architecture as the teacher network, and knowledge distillation aligns the candidate architectures with F1 score of range 0.832-0.838. Then, the weight grafting in Eq. (7) further improves the best F1 score by 0.3%.

6. Conclusion

In this work, we collect a new short video shot boundary detection dataset, named SHOT, which is a quite different scenario from conventional long video based shot boundary detection. The SHOT can significantly accelerate the development and evaluation of various short video based applications, *e.g.*, intelligent creation, and video scene segmentation and understanding. Leveraging this new asset, we propose to optimize the model design for the task of video shot boundary detection, by conducting neural architecture search in a search space encapsulating various advanced 3D ConvNets and Transformers. AutoShot surpasses previous best shot boundary detection method by 4.2% and 3.5% based on the F1 and precision scores, respectively. We further validate the generalizability of the searched optimal architecture on ClipShots, BBC and RAI. Experimental results demonstrate that, AutoShot has a good generalizability and outperforms previous state-of-the-art approaches on the three existing public datasets.

References

[1] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. A deep siamese network for scene detection in broadcast videos. In *Proceedings of the 23rd ACM International Conference on Multimedia*, pages 1199–1202, 2015. 1, 2, 3, 7

[2] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. Shot and scene detection via hierarchical clustering for reusing broadcast video. In *Proceedings of International Conference on Computer Analysis of Images and Patterns*, pages 801–811. Springer, 2015. 1, 2, 3, 7

[3] Brandon Castellano. Pyscenedetect. http://www.bcastell.com/projects/PySceneDetect, 2022. 3, 7

[4] Saptarshi Chakraborty, Alok Singh, and Dalton Meitei Thounaojam. A novel bifold-stage shot boundary detection algorithm: invariant to motion and illumination. *The Visual Computer*, 38(2):445–456, 2022. 2

[5] Vasileios Chasanis, Aristidis Likas, and Nikolaos Galatsanos. Simultaneous detection of abrupt cuts and dissolves in videos using support vector machines. *Pattern Recognition Letters*, 30(1):55–65, 2009. 6

[6] Shixing Chen, Xiaohan Nie, David Fan, Dongqing Zhang, Vimal Bhat, and Raffay Hamid. Shot contrastive self-supervised learning for scene boundary detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9796–9805, 2021. 1, 3

[7] Yuantao Feng, Shiqi Yu, Hanyang Peng, Yan-Ran Li, and Jianguo Zhang. Detect faces efficiently: A survey and evaluations. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 4(1):1–18, 2021. 4

[8] Jacob R Gardner, Matt J Kusner, Zhixiang Eddie Xu, Kilian Q Weinberger, and John P Cunningham. Bayesian optimization with inequality constraints. In *Proceedings of International Conference on Machine Learning*, volume 2014, pages 937–945, 2014. 6

[9] Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, and Jian Sun. Single path one-shot neural architecture search with uniform sampling. In *Proceedings of European Conference on Computer Vision*, pages 544–560. Springer, 2020. 2, 3, 6

[10] Michael Gygli. Ridiculously fast shot boundary detection with fully convolutional neural networks. In *Proceedings of 2018 International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–4. IEEE, 2018. 3

[11] Ahmed Hassanien, Mohamed Elgharib, Ahmed Selim, Sung-Ho Bae, Mohamed Hefeeda, and Wojciech Matusik. Large-scale, fast and accurate shot boundary detection through spatio-temporal convolutional neural networks. *arXiv preprint arXiv:1705.03281*, 2017. 1, 3, 8

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1, 3

[13] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Proceedings of 28th International Conference on Neural Information Processing Systems Deep Learning Workshop*, 2014. 6

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of International Conference on Machine Learning*, pages 448–456. PMLR, 2015. 1, 3

[15] Qing-Yuan Jiang, Yi He, Gen Li, Jian Lin, Lei Li, and Wu-Jun Li. Svd: A large-scale short video dataset for near-duplicate video retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5281–5289, 2019. 3

[16] Xuekun Jiang, Libiao Jin, Anyi Rao, Linning Xu, and Dahua Lin. Jointly learning the attributes and composition of shots for boundary detection in videos. *IEEE Transactions on Multimedia*, 2021. 2

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105, 2012. 6

[18] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 780–787, 2014. 2

[19] Yang Li, Yu Shen, Wentao Zhang, Yuanwei Chen, Huaijun Jiang, Mingchao Liu, Jiawei Jiang, Jinyang Gao, Wentao Wu, Zhi Yang, et al. Openbox: A generalized black-box optimization service. In *Proceedings of 27th SIGKDD Conference on Knowledge Discovery and Data Mining (KDD-2021)*, 2021. 7

[20] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *Proceedings of International Conference on Learning Representations*, 2018. 6

[21] Jakub Lokoč, Gregor Kovalčik, Tomáš Souček, Jaroslav Moravec, and Přemysl Čech. A framework for effective known-item search in video. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1777–1785, 2019. 1, 3, 8

[22] Fanxu Meng, Hao Cheng, Ke Li, Zhixin Xu, Rongrong Ji, Xing Sun, and Guangming Lu. Filter grafting for deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6599–6607, 2020. 6, 7

[23] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of International Conference on Machine Learning*, 2010. 6

[24] Hanyang Peng and Shiqi Yu. A systematic iou-related method: Beyond simplified regression for better localization. *IEEE Transactions on Image Processing*, 30:5032–5044, 2021. 4

[25] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5533–5541, 2017. 2, 3, 5

[26] Anyi Rao, Linning Xu, Yu Xiong, Guodong Xu, Qingqiu Huang, Bolei Zhou, and Dahua Lin. A local-to-global approach to multi-modal movie scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10146–10155, 2020. 1, 3

[27] BS Rashmi and HS Nagendraswamy. Video shot boundary detection using block based cumulative approach. *Multimedia Tools and Applications*, 80(1):641–664, 2021. 2

[28] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015. 2, 6

[29] Mike Zheng Shou, Stan W Lei, Weiyao Wang, Deepti Ghadiyaram, and Matt Feiszli. Generic event boundary detection: A benchmark for event segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 1, 3

[30] Tomáš Souček and Jakub Lokoč. Transnet v2: An effective deep network architecture for fast shot transition detection. *arXiv preprint arXiv:2008.04838*, 2020. 1, 3, 4, 5, 6, 7, 8

[31] Sebastian Stein and Stephen J McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 729–738, 2013. 2

[32] Shitao Tang, Litong Feng, Zhanghui Kuang, Yimin Chen, and Wei Zhang. Fast video shot transition localization with deep structured models. In *Proceedings of Asian Conference on Computer Vision*, pages 577–592. Springer, 2018. 1, 2, 3, 7, 8

[33] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1207–1216, 2019. 2

[34] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4489–4497, 2015. 3

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 2, 3, 5

[36] Zhenzhi Wang, Zhimin Li, Liyu Wu, Jiangfeng Xiong, and Qinglin Lu. Overview of tencent multi-modal ads video understanding. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4725–4729, 2021. 1, 3

[37] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 305–321, 2018. 1, 3

[38] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *Proceedings of International Conference on Learning Representations*, 2016. 1, 3

[39] Xuefan Zha, Wentao Zhu, Lv Xun, Sen Yang, and Ji Liu. Shifted chunk transformer for spatio-temporal representational learning. *Advances in Neural Information Processing Systems*, 34:11384–11396, 2021. 1, 2

[40] Wentao Zhu, Yufang Huang, Liang Zeng, Xuming Chen, Yong Liu, Zhen Qian, Nan Du, Wei Fan, and Xiaohui Xie. Anatomynet: deep learning for fast and fully automated whole-volume segmentation of head and neck anatomy. *Medical physics*, 46(2):576–589, 2019. 2

[41] Wentao Zhu, Tianlong Kong, Shun Lu, Jixiang Li, Dawei Zhang, Feng Deng, Xiaorui Wang, Sen Yang, and Ji Liu. Speechnas: Towards better trade-off between latency and accuracy for large-scale speaker verification. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1102–1109. IEEE, 2021. 2

[42] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016. 1

[43] Wentao Zhu, Chaochun Liu, Wei Fan, and Xiaohui Xie. Deeplung: Deep 3d dual path nets for automated pulmonary nodule detection and classification. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 673–681. IEEE, 2018. 2

[44] Wentao Zhu, Jingya Liu, and Yufang Huang. Hnssl: Hard negative-based self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2023. 3

[45] Wentao Zhu, Yeeleng S Vang, Yufang Huang, and Xiaohui Xie. Deepem: Deep 3d convnets with em for weakly supervised pulmonary nodule detection. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11*, pages 812–820. Springer, 2018. 2