Video Multitask Transformer Network

Hongje Seong Junhyuk Hyun Euntai Kim*
Yonsei University, Seoul, Korea
{hjseong, jhhyun, etkim}@yonsei.ac.kr

Abstract

In this paper, we propose the Multitask Transformer Network for multitasking on untrimmed video. To analyze the untrimmed video, it needs to capture important frame and region in the spatio-temporal domain. Therefore, we utilize the Transformer Network, which can capture the useful features from CNN representations through an attention mechanism. Motivated by the Action Transformer Network, which is a repurposed model of the Transformer for video, we modified the concept of query which was specialized only for action recognition on the trimmed video to fit the untrimmed video. In addition, we modified the structure of the Transformer unit to the pre-activation structure for identity mapping on residual connections. We also utilize the class conversion matrix (CCM), one of the feature fusion methods, to share the information of different tasks. Combining our Transformer structure and CCM, the Multitask Transformer Network is proposed for multitasking on untrimmed video. Eventually, our model evaluated on CoVieW 2019, and we enhanced the performance through post-processing based on prediction results that suitable to the CoVieW 2019 evaluation metric. In CoVieW 2019 challenge, we placed fourth on final rank while first on scene and action score.

1. Introduction

Analyzing and understanding the untrimmed video is a fundamental problem for real-world artificial intelligence. Recently, deep CNN-based methods achieve state-of-the-art performance on image recognition such as object recognition [34, 21, 39, 13, 18], scene recognition [48, 36], object detection [28, 33], semantic segmentation [6, 47], and object tracking [8, 27]. Since video is a sequence of images, many previous video recognition approaches use CNN representations.

The untrimmed video contains a lot of useless frames, so we do not need to analyze all of the CNN features for every frame. To filter out useless features, we utilized the Action Transformer Network [9], which can capture the interested actioness region through an attention mechanism by repurposing the query, key, and value (QKV) concept of the Transformer Network [44]. We also modified the query concept and the Transformer unit structure of the Action Transformer Network to capture the important features in untrimmed videos.

For multitask learning, we expect that performance would be improved by sharing the information when jointly training the different categories of labels. To do this, we fuse the features of different tasks. Many existing methods use concatenation or element-wise sum to perform feature fusion [38, 4]. However, if we perform feature fusion using a class conversion matrix (CCM) [35], we likely to get improved performance than conventional feature fusion methods since it adapted the domains somewhat between two different features. To capture important frames in an untrimmed video with sharing the information in different tasks, we propose the Multitask Transformer utilizing the CCM and Action Transformer based on CNN representations. The overall architecture of our proposed Multitask Transformer Network is shown in Fig. 1.

Finally, we used CoVieW 2019[1] to train and evaluate our model. CoVieW 2019 provides untrimmed videos with three labels that scene, action, and importance scores for a single video segment. To evaluate on CoVieW 2019, it requires the selection of the top-6 segments in the video based on the importance score. To improve the performance on CoVieW 2019 evaluation metric, we propose a method of importance score recalibration through prediction-based post-processing. In CoVieW 2019 challenge, we took fourth place on final rank while first place on scene and action score without the use of model ensemble and optical flow.

2. Related work

In this section, we briefly review some related studies about two topics: (1) feature aggregation methods, and (2) feature fusion methods.

*Corresponding author.

[1] http://cvlab.hanyang.ac.kr/coview2019.

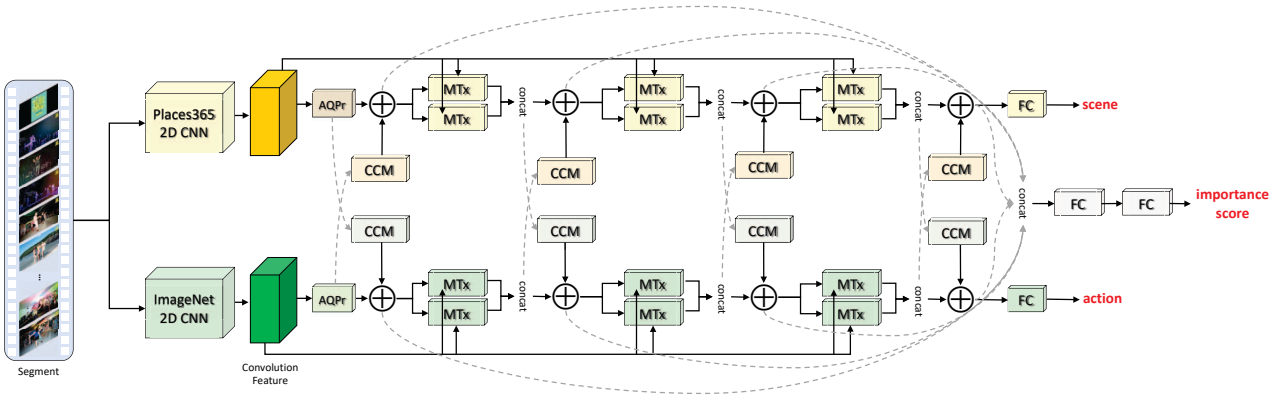


Figure 1. The overall architecture with our Multitask Transformer Network for CoVieW 2019. Places365-2D CNN and ImageNet-2D CNN extract scene and object features, respectively. The attentional query processor (AQPr) generates a query. The Multitask Transformer Network consists of a stack of Multitask Transformer (MTx) units, which generates the features to be classified for each task, and CCM [35] for fuse the features. When training, the solid black line propagates the gradient in backward, but the dotted gray line does not propagate the gradient. The structure of AQPr and MTx explained in Section 3.2.

2.1. Feature aggregation for video recognition

Recently, many video recognition approaches based on the CNN representations [38, 7, 20, 42, 32] rather than the hand-crafted feature [24, 25, 45]. The CNN-based methods focused on feature aggregation methods to generating a new feature for well-recognizing video. The simplest method is that generating a feature via a global average pooling (GAP) [12, 3], but it cannot catch the important features that appear only in certain frames or regions. Therefore, to catch the important feature, the attention module is often used [37, 31]. The Action Transformer Network [9] repurposes the Transformer Network [44], which was used in natural language processing through self-attention mechanism with the QKV concept, and the QKV concept was modified to suit the action recognition. Since video is a sequence of images on a time domain, the recurrent neural network (RNN) such as long short-term memory (LSTM) [15] or gated recurrent unit (GRU) [5] is also utilized for aggregating CNN features [3]. Another way, aggregating features using NetVLAD [1], which used for place recognition, performed well in action recognition and video classification [10, 29]. Except for RNN methods, the methods introduced in this section are orderless aggregation methods, and we utilize the Action Transformer to aggregating features of untrimmed video.

2.2. Feature fusion

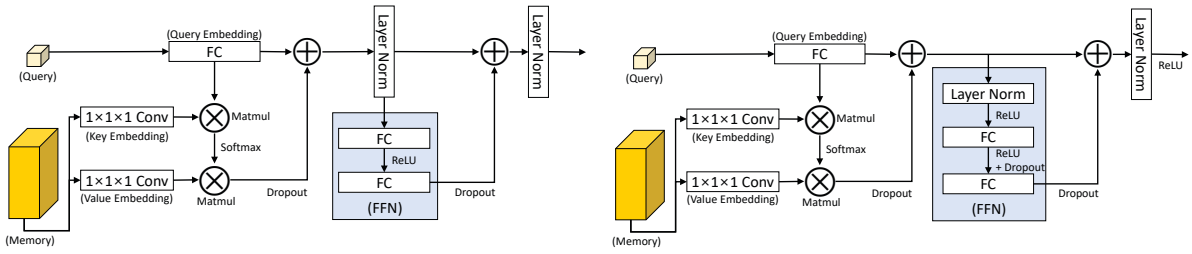
In multitask learning, the fusion of features which extracted for analyzing each task is often used to boosting performance [23, 16]. The feature fusion is also used in singletask learning; utilizing object information for scene recognition [4, 36] and utilizing optical flow information for action recognition [38, 7]. Most of these conventional approaches use element-wise summation, element-wise multiplication, or concatenation when performing feature fusion. The class conversion matrix (CCM) [35] likely achieves better performance than the existing fusion method by transforming the input feature to match the domain to be fused. In our model, feature fusion is used to share the information of each task, we adopt CCM for fusion method.

3. Methods

In this section, we introduce our methods for recognizing the scene and action simultaneously on untrimmed videos. We describe the feature extraction methods based on CNN in Section 3.1, and the explanation of our Multitask Transformer Network is followed in Sections 3.2, and 3.3. Then, we describe our importance score regression method in Section 3.4 to evaluate our model on CoVieW 2019. The overall our proposed architecture is given in Fig. 1.

3.1. Feature extraction

Since training CNN from scratch on the CoVieW 2019 dataset caused serious overfitting problem, so we obtained features of video using pre-trained CNN, the experimental analysis is given in Section 4.4. To obtaining CNN representations for scene recognition, we used features extracted from `conv5_x` layer (denotes in [13]) of Places365-2D CNN, which pre-trained on Places 365 dataset [48]. Similarly, we used ImageNet-2D CNN, which pre-trained on ImageNet [34], to extract features for action recognition. Previous studies generally extract features using 3D CNN from RGB images or use additional optical flow when using only 2D CNN. However, there are some cons, optical flow requires much pre-computation, and 3D CNN requires a lot of memory, it leads difficult to use deep architecture. In view of [26], the action recognition tasks such as [22, 40]



(a) Action Transformer (Tx) unit [9]

(b) Our Multitask Transformer (MTx) unit

Figure 2. The structure of (a) the Action Transformer (Tx) unit [9] and (b) our Multitask Transformer (MTx) unit. The existing Tx unit structure has a normalization module in the residual connection of the query. We modified the position of this normalization module to the pre-activation [14] of the Feed-forward Network (FFN).

can recognize an action even if only a single frame is given by understanding spatial context related to action. Interestingly, we also found that ImageNet-2D CNN achieve better performances than Kinetics-3D CNN [3] for action recognition. Therefore, we use 2D CNN for action feature extraction. In the case of scene recognition, it has limited to generates a pre-trained 3D scene CNN because there was no large amount of videos for scene recognition. In addition, since the scene does not change dynamically on time, we expected to be able to recognize the scene with 2D CNN representation. Therefore, we also use 2D CNN for scene recognition. The comparison experiment between 2D CNN and 3D CNN for action recognition is given in Section 4.4.

3.2. Spatio-temporal attention module

The untrimmed videos include a lot of useless frames to recognizing the scene or action, so we wanted to capture only the important features in spatio-temporal domain. Therefore, we utilized the Action Transformer Network [9], which is a modified for video action recognition by revising the QKV concepts of the Transformer Network [44] which can capture the important feature through an attention mechanism. The Action Transformer set the query to a RoIPool-ed feature for the actioning person box, which obtained by supervised learning. However, CoVieW 2019 dataset provides only categories of scene and action labels without a bounding box of sceness or actioness, so we needed to modify the concept of the query used in Action Transformer. Therefore, we generate a query through the attentional query processor (AQPr), which learned weakly supervised to select only useful features in the untrimmed video. For each spatio-temporal feature $X_{t,h,w}$, the AQPr generates attention map $M_{t,h,w}$ and a refined feature $Y^{attention}$ as follows:

$$M_{t,h,w} = \sigma\left(\text{InstanceNorm}\left(W^{Attention} X_{t,h,w}\right)\right), \quad (1)$$

$$Y^{attention} = \sum_{t,h,w} \frac{M_{t,h,w} X_{t,h,w}}{\sum_{t,h,w} M_{t,h,w}}, \quad (2)$$

where $\sigma(\cdot)$ denotes a sigmoid function, $W^{Attention}$ is trainable parameter and InstanceNorm [43] also contains trainable parameters, and the attention map $M_{t,h,w}$ is a scalar. To generate the attention map $M_{t,h,w}$, we can replace InstanceNorm with trainable scalar value (bias) or BatchNorm [19], but experimentally InstanceNorm has the best performance for Multitask Transformer Network. The $Y^{attention}$ was used as the query of our Multitask Transformer (MTx) unit, as shown in Fig. 1.

The features of key K and value V for our MTx unit are generated through a simple linear projection of the CNN representations without aggregation. Then, the query, and memory are fed into the MTx unit, and the structure of MTx unit is given in Fig. 2(b). We modified the Action Transformer (Tx) unit [9] by replacing with identity residual connection and adding some non-linearity. As shown in Fig. 2(a), the existing Tx unit performs the normalization twice on the residual connection. However, since adding any module into the residual connection may cause performance degradation by optimization issues, we modified the structure of the Feed-forward Network (FFN) to a pre-activation type structure [14]. Furthermore, we added some non-linearity by adding the ReLU activation function after each LayerNorm [2].

The remaining architecture essentially follows the Action Transformer Network [9]. The linear projected query Q, key K, and value V all have the same channel dimension as D. We used the D as 128. After matrix multiplication of Q and K, we normalize it by dividing to $\sqrt{D}$ before applying softmax as in [44]. Then, the V is weighted summed by matrix multiplication with softmax attention, and dropout [41] of 0.1 is followed. After updating the query with a weighted summed value, we update the query again through our pre-activated FFN. Finally, the LayerNorm and ReLU are followed.

3.3. Feature fusion method

To take advantage of multitask learning, we performed feature fusion by sharing the information of different tasks.

Many of the existing methods use concatenation, sum or multiplication for feature fusion, but we opt for CCM [35]. If CCM is used for feature fusion, the dimension of output fused feature is not changed by using a sum operator rather than concatenation, and since the non-linearity is increased by using ReLU in CCM, the performance likely increases. With queries Q_α and Q_β that extracted from each module for task α and task β, respectively, the fused queries Q'_α for task α is calculated via CCM as follows:

$$Q'_\alpha = Q_\alpha + \text{ReLU}\left(W_\alpha Q_\beta + b_\alpha\right) \tag{3}$$

where W_α and b_α are trainable class conversion matrices for task α. We applied the CCM to every query of the MTx units and applied CCM to both scene and action queries. The detailed position of the CCM is given in Fig. 1. In addition, the CCM does not backpropagate the gradient when training to increase the expertise of each task. If the CCM backpropagates the gradient, performance degradation has occurred, and the experimental results are given in Section 4.4.

3.4. Importance score regression

To evaluate on CoVieW 2019 challenge, we need to regress the importance score for each segment. An importance score is an indicator of how important a given segment is in the video, and it given in real number bounded from 0 to 2. Based on the importance score, six segments with a high importance score be selected from a video, and the scene and action hamming scores for only these six segments are measured.

To regress importance score, we concatenate all queries and fed it into importance score regressor. The importance score regressor implemented as a 2-layer MLP with a half dimension of hidden layer, ReLU activation, and dropout of 0.1. We did not propagate the gradient of the importance score regression error to the Multitask Transformer and the feature extractor, because it severely interferes with the training of the scene and action. Since the importance score is bounded, we could use cross-entropy loss for calculate the error of importance score. Thus, although the importance score prediction is not a classification task, we opt loss function for the binary cross-entropy with sigmoid activation function rather using the mean squared error (MSE). We also tried to extract the feature from the whole video rather than a segment for accurate prediction of importance score value. In this case, because of insufficient memory, we used only a single 2D CNN and gathered video features from only one image for each segment. However, the performance of video-level model is lower than the segment-level network, the experimental results are given in Section 4.4.

The labels of importance score seem to highly subjective and very noisy. We tried to learn the importance score but found that even we cannot overfit the importance score to training data. In addition, scene and action accuracy for selected top-6 segments based on importance score regression results were significantly degraded than segment-level accuracy. Therefore, we select important segments using more clear and accurate results, scene and action. We determined the reliability of the prediction result for a segment by obtaining the maximum value of predictions applied softmax. The reliability was extracted separately from the scene and action, and then multiplied by the regressed importance score and used it as the final predicted importance score. For each n-th segment of the given video, let P_n^{scene} and P_n^{action} be the vector of prediction scores for each class of scene and action, respectively, and $P_n^{importance}$ be the scalar of importance score regression result. Then, the new importance score $P_n^{new\_importance}$ using the our proposed prediction-based post-processing is calculated as follows:

$$\begin{aligned} P_n^{new\_importance} = &\max\left(\text{softmax}\left(P_n^{scene}\right)\right) \\ &\times \max\left(\text{softmax}\left(P_n^{action}\right)\right) \\ &\times \text{sigmoid}\left(P_n^{importance}\right). \end{aligned} \tag{4}$$

If we select the top-6 important segments based on the calculated $P_n^{new\_importance}$, the importance score accuracy becomes slightly degraded, but the scene and action hamming score are significantly improved. The experimental results are given in Section 4.3.

4. Experiments

In this section, we describe the experimental results of the Multitask Transformer Network on the CoVieW 2019 dataset. We describe contents of the CoVieW 2019 dataset in Section 4.1, and we describe the training settings in Section 4.2. In Section 4.3, we show the experimental results on CoVieW 2019, and we conduct the ablation studies in Sections 4.4.

4.1. CoView 2019 dataset

CoVieW 2019 dataset provides untrimmed videos with the scene, action, and importance score labels for each 5-second segments. The total number of the provided video is 1500, of which 1200 are training data and the remaining 300 are test data, which not include labels. Since there is no validation set, we arbitrarily split it into training 1000 videos and validation 200 videos with balanced the number of labels between the training set and validation set as much as possible. The scene and action contain 78 and 99 categories, respectively. The importance score indicates how important each segment is, compared to other segments from the same video. The CoVieW 2019 challenge uses the hamming score of the scene and action, and the importance score accuracy. The hamming score of the scene and action is calculated only for 6 segments per video, which selected based on the

Feature aggregation method	Segment-level results		CoVieW 2019 evaluation metric					
			Based on IS regression results			After post-processing		
	Scene	Action	Scene	Action	IS	Scene	Action	IS
GAP	55.879	52.799	52.250	45.667	**0.793**	60.167	62.083	0.745
Only AQPr	56.045	53.539	53.583	44.500	0.787	61.250	60.500	0.753
GAP + Concat Fusion	55.603	55.018	54.833	45.583	0.780	61.667	60.750	0.763
GAP + CCM Fusion	55.570	**55.073**	55.083	45.417	0.783	62.000	63.167	0.759
Multitask Transformer	**56.586**	54.654	53.750	44.500	0.789	**63.167**	**66.000**	0.732

Table 1. Experimental validation results based on segment-level and CoVieW 2019 evaluation metric using 2D ResNet-18. For scene and action, top-1 accuracy was measured. IS denotes the importance score, and accuracy of the importance score measured by CoVieW 2019 metric.

Input size ($T \times H \times W$)	Segment-level results		CoVieW 2019 evaluation metric					
			Based on IS regression results			After post-processing		
	Scene	Action	Scene	Action	IS	Scene	Action	IS
$32 \times 128 \times 128$	58.258	55.641	56.833	46.833	0.793	64.333	66.250	0.743
$32 \times 224 \times 224$	60.086	**57.812**	58.583	51.500	0.797	65.333	**67.750**	0.745
$32 \times 224 \times 224$, ten-crop	60.241	57.282	60.833	51.333	**0.798**	**67.833**	67.667	0.741
All $\times 224 \times 224$, ten-crop	**60.351**	57.315	61.167	51.083	0.797	67.667	67.417	0.742

Table 2. The validation results of the Multitask Transformer Network based on CoVieW 2019 evaluation metric using 2D SE-ResNeXt-101. The results of the three bottom rows are the same model trained with $32 \times 224 \times 224$ input size, and only testing methods are different.

importance score. To calculate the importance score accuracy, the average of ground truth importance score of the selected six segments is divided by the maximum value of six importance scores that the video can have.

4.2. Implementation details

We optimize our model using distributed synchronous stochastic gradient descent with Nesterov momentum of 0.9 [11, 30]. The learning rate initially set to 0.001 for a batch size of 256, and divide it by 10 after the validation loss saturates. Here, we use the various batch size between 32 and 256, and we adjust the learning rate by applying the linear scaling rule [11] to obtaining the same performance for different batch sizes. We also use gradual warmup as in [11], and assign a weight decay of 1e-4. We select random 32 frames sequence from the given segment and random crop the frame to 128×128 size. To boosting performance, only a final model uses 224×224 input size of the image. Frames were selected temporally-ordered for 3D CNN experiments and temporally-orderless for 2D CNN experiments. Since the videos provided in the CoVieW 2019 have different frame rates, we sampled the frames with a fixed frame rate of 25. We used ResNet-18 [13] and SE-ResNeXt-101 [46, 17] as backbone networks, random initialization for only the last fully-connected layer, and all convolution layers initialized to weights of pre-trained on Places 365 [48] and ImageNet [34], as explained in Section 3.1.

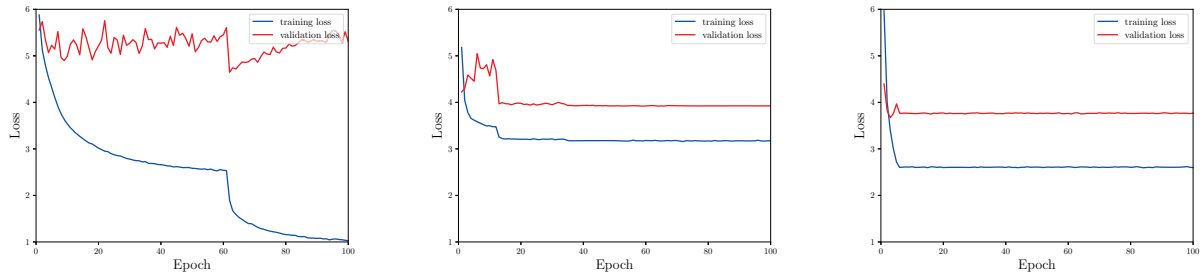
4.3. CoVieW 2019 results

Experimental results based on the CoVieW 2019 metric for our models are given in Table 1. For this experiment, only the 2D CNNs were used as the backbone architecture, and the weights of 2D CNNs before `conv5_x` blocks were frozen. In Table 1, we show segment-level results and CoVieW 2019 evaluation metric results, and it has two types of results based on the selection methods of important segments; based on regressed importance score, and our prediction based importance score post-processing methods (explained in Section 3.4). Experimental results show that when selecting important segments based on the importance score regression result, it is difficult to find any pattern of scene and action performance according to the feature aggregation method. This indicates that the learning of the importance score is very difficult. Therefore, applying our prediction based selection method with scene and action classification results, we can find and analyze some patterns of results, and scene and action performances are significantly improved. Aggregating features with only a AQPr has a slightly better scene performance, but action performance becomes slightly degraded. The performance was improved by performing the feature fusion of scene and action using concat or CCM, and CCM has more improved performance than concat fusion methods. The Multitask Transformer achieves the best performance than other aggregation methods.

Furthermore, we also train the Multitask Transformer using the SE-ResNeXt-101 as the deeper backbone network

Submission ID	summary score	rank	scene & action score	rank	Final rank (rank sum)
ID 13	**0.8512**	1	0.7325	3	1 (4)
ID 3	0.8343	3	0.8036	2	2 (5)
ID 19	0.8409	2	0.7294	4	3 (6)
ID 5 (ours)	0.7594	7	**0.8114**	1	4 (8)
ID 15	0.8151	4	0.3872	5	5 (9)
ID 16	0.7985	5	0.3831	6	6 (11)
ID 14	0.7696	6	0.0494	7	7 (13)

Table 3. Performance comparison of different methods on CoVieW 2019 testset. In CoVieW 2019 challenge, scene & action score is measured by the top-5 hamming score.



(a) From scratch (b) No finetune (case A) (c) Finetune `conv5_x` as in [12] (case B)

Figure 3. Training and validation loss curves for the different finetuning backpropagated positions of feature extractor using GAP aggregation method, which is a basic feature aggregation method. (a) is a result of training from scratch. (b) is a result of freezing pre-trained weights and training only a last fully-connected layer for classification. (c) is a result of finetuning from the `conv5_x` block as in [12].

to boosting performance for CoVieW 2019 challenge, and the results are given in Table 2. The ten-crop methods applied only on the spatial domain, and we used the standard ten-crop method as used in [21]. In the case of 'All' at the input size of T, all the frames in the segment are used as an input. Since all frames have different frame rates, it cannot be fixed to one value. Both the result using ten-crop and the result of using 'All' as temporal input are trained with a $32 \times 224 \times 224$ input size, and they differ only the test method. Interestingly, with ten-crop testing, the performance of the scene is improved, but the performance of the action is slightly decreased. Furthermore, using the temporal input size as all frames did not much helpful for improving performance.

Finally, we submitted the results for CoVieW 2019 challenge using the Multitask Transformer Network of the $32 \times 224 \times 224$ input size with ten-crop testing with the backbone architecture of the SE-ResNeXt-101 model. As a result, our model took 4th place in CoVieW 2019 challenge, and 1st place on scene & action score as shown in Table 3. Our Multitask Transformer Network uses only segment-level inputs, so it is weak for regressing importance score, but it can captures important frames on untrimmed video well, so our model achieves the best performance on scene and action classification task.

CNN - Finetuning level	Scene	Action
2D CNN - A	53.826	52.302
2D CNN - B	**55.879**	**52.799**
3D CNN - A	-	45.876
3D CNN - B	-	51.899

Table 4. Comparison of 2D CNN and 3D CNN on the various finetuning level using ResNet-18 architecture with GAP aggregation method.

4.4. Ablation studies

To verify our Multitask Transformer Network on various perspective, we conducted several ablation studies on: backbone CNN architecture and finetuning, normalization methods in AQPr, feature fusion method, MTx unit, and importance score regression method. All ablation experiments used segment-level scenes and action accuracies, which are a more precise comparison metric to compare the model than evaluate the accuracies only on selected important segments in our view.

Backbone CNN architecture and finetuning. Before decided to use pre-trained models, we train the whole network from scratch using ResNet-18, and the curves of training and validation error are given in Fig. 3. For this experiment, we used 2D CNN for the scene, 3D CNN for

Normalization method	Scene	Action
None	56.133	53.704
None, bias	56.491	53.864
Batch Normalization	55.957	54.025
Instance Normalizaion	**56.586**	**54.654**

Table 5. Comparison of normalization methods in AQPr for generating a query of the Multitask Transformer Network using 2D ResNet-18. The scene and action accuracies are computed on the segment-level.

Feature fusion method	Scene	Action
Concat	55.603	55.018
CCM - backpropagated	**55.708**	53.500
CCM - no backpropagated	55.570	**55.073**

Table 6. Comparison of feature fusion methods using 2D ResNet-18 with GAP aggregation method. The scene and action accuracies are computed on the segment-level.

the action, and Places365-2D CNN and Kinetics-3D CNN for the pre-trained models, respectively, and use the GAP aggregation method. As shown in Fig. 3(a), which is the result of the training from scratch, the training error is consistently reduced, but the validation error can be seen to never converge. Therefore, we used a well-trained pre-trained model on large scale datasets, and we determined to finetuning CNN or not by analyzing experimental results in Figs. 3(b) and 3(c). This experiment shows that finetuning `conv5_x` as in [12] has better performance. In addition, we tried to replace the Kinetics-3D CNN for action recognition with ImageNet-2D CNN, and the results are shown in Table 4. As shown in Table 4, ImageNet-2D CNN has better performance than Kinetics-3D CNN, and the finetuning `conv5_x` block has best performance. Therefore, all our experiments used Places365-2D CNN for scene recognition and ImageNet-2D CNN for action recognition, and we trained from the `conv5_x` block of pre-trained CNN with feature aggregation module.

Normalization method in AQPr. To generate query feature of the Multitask Transformer Network, we use AQPr, and it uses InstanceNorm (in Equation 1). We can replace this InstanceNorm with BatchNorm or bias without using any normalization method. The experimental results are shown in Table 5. We can find that the query generated through InstanceNorm is well-suited with the Multitask Transformer Network. This means that normalization helps to prevent deactivating or activating too many features and it is better to determine the activated features without affected by other segments.

Feature fusion method. To determine the feature fusion method, we compare the concatenation method and the CCM method, and the results are given in Table 6. The simple concatenation method performed very well, but the feature fusion method through CCM achieves better performance. Interestingly, the experimental results show the CCM, which does not backpropagate the gradient during training, has much better performance for action recognition. It suggests that having the expertise for each task without disturbed from other tasks would be better for multitask learning.

Transformer unit	Scene	Action
Tx [9]	56.111	54.278
MTx (ours)	**56.586**	**54.654**

Table 7. The experimental results of replacing our proposed Multitask Transformer (MTx) unit with the existing Action Transformer (Tx) unit. The scene and action accuracies are computed on the segment-level.

Training labels	Method	IS
only IS	Video-level	0.760
	Segment-level	0.776
IS+scene+action	Segment-level	**0.793**

Table 8. Comparison of training methods for regressing importance score using 2D ResNet-18. The scene and action accuracies are computed on the segment-level.

MTx unit. To verify the effectiveness of our proposed MTx unit, we replace the MTx units of our Multitask Transformer Network with the existing Action Transformer (Tx) units, and the results are given in Table 7. As described in Section 3.2, our MTx unit is able to compensate for the disadvantages of the existing Tx unit to achieve better performance.

Importance score regression method. The ablation study of the importance score training is given in Table 8. Interestingly, learning the scene and action jointly achieves the improved importance score performance than learning only an importance score. However, if we backpropagate the gradient of importance score error to feature extraction module and feature aggregation module, the performance of the scene and action degraded dramatically compared to the improvement of the accuracy of importance score. Therefore, we train the importance score jointly scene and action, but the gradient of the importance score error only propagated to the last two fully-connected that unaffected on the scene and action prediction.

5. Conclusions

In this paper, we proposed the Multitask Transformer for comprehensive video understanding in the wild. Our proposed model consists of attention module and feature fusion module for jointly multitask learning. We also applied post-processing to the regression result of importance score to solve the problem of noisy importance score labels for

CoVieW 2019 evaluation metric. Finally, we achieve the high accuracy of the scene and action, even though the accuracy of the importance score was slightly low.

Acknowledgment

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT (NRF-2017M3C4A7069370).

References

[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, pages 5297–5307, 2016. 2

[2] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3

[3] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 2, 3

[4] X. Cheng, J. Lu, J. Feng, B. Yuan, and J. Zhou. Scene recognition with objectness. *Pattern Recognition*, 74:474–487, 2018. 1, 2

[5] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014. 2

[6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016. 1

[7] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, pages 1933–1941, 2016. 2

[8] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361. IEEE, 2012. 1

[9] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman. Video action transformer network. In *CVPR*, pages 244–253, 2019. 1, 2, 3, 7

[10] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *CVPR*, pages 971–980, 2017. 2

[11] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch sgd: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 5

[12] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, pages 6546–6555, 2018. 2, 6, 7

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 2, 5

[14] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 3

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2

[16] Y. W. Hong, H. Kim, and H. Byun. Multi-task joint learning for videos in the wild. In *ACM MM Workshop*, 2018. 2

[17] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *CVPR*, June 2018. 5

[18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, July 2017. 1

[19] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 3

[20] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012. 2

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 1, 6

[22] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, pages 2556–2563. IEEE, 2011. 2

[23] H. Kwon, S. Kwak, and M. Cho. Video understanding via convolutional temporal pooling network and multimodal feature fusion. In *ACM MM Workshop*, 2018. 2

[24] I. Laptev, B. Caputo, et al. Recognizing human actions: a local svm approach. In *ICPR*. IEEE, 2004. 2

[25] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *ICCV*, 2008. 2

[26] M. Lee, S. Lee, S. Son, G. Park, and N. Kwak. Motion feature network: Fixed motion filter for action recognition. In *ECCV*, pages 387–403, 2018. 2

[27] S. Lee and E. Kim. Multiple object tracking via feature pyramid siamese networks. *IEEE Access*, 7:8181–8194, 2018. 1

[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 1

[29] A. Miech, I. Laptev, and J. Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017. 2

[30] Y. Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018. 5

[31] J. Park, S. Jeon, S. Kim, J. Lee, S. Kim, and K. Sohn. Learning to detect, associate, and recognize human actions and surrounding scenes in untrimmed videos. In *ACM MM Workshop*, 2018. 2

[32] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *CVPR*, pages 5533–5541, 2017. 2

[33] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, pages 7263–7271, 2017. 1

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 1, 2, 5

[35] H. Seong, J. Hyun, H. Chang, S. Lee, S. Woo, and E. Kim. Scene recognition via object-to-scene class conversion: end-to-end training. In *IJCNN*, July 2019. 1, 2, 4

[36] H. Seong, J. Hyun, and E. Kim. Fosnet: An end-to-end trainable deep neural network for scene recognition. *arXiv preprint arXiv:1907.07570*, 2019. 1, 2

[37] H. Seong, J. Hyun, S. Lee, S. Woo, H. Chang, and E. Kim. New feature-level video classification via temporal attention model. In *ACM MM Workshop*. ACM, 2018. 2

[38] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014. 1, 2

[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[40] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2

[41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 3

[42] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015. 2

[43] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 3

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. 1, 2, 3

[45] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, pages 3551–3558, 2013. 2

[46] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 1492–1500, 2017. 5

[47] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *ECCV*, pages 325–341, 2018. 1

[48] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. 1, 2, 5