# NII_UIT AT TRECVID 2020

Duy-Dinh Le[1], Hung-Quoc Vo[1], Dung-Minh Nguyen, Tien-Van Do[1],
Thinh-Le-Gia Pham[1], Tri-Le-Minh Vo[1], Thua-Ngoc Nguyen[1],
Vinh-Tiep Nguyen[1], Thanh-Duc Ngo[1],
Zheng Wang[2], Shin'ichi Satoh[2]
[1] University of Information Technology, VNU-HCMC, Vietnam
[2] National Institute of Informatics, Japan

# Chapter 1

# Instance Search (INS)

## 1.1 Abstract

Following last year, TRECVID INS 2020 continues to tackle the problem of searching for specific person doing specific action. This year, we use the same person search system as in 2019 and only focus on improving the action search. Precisely, this work concentrate on solving the case when both the target person and the desired action appear in the shot but that action is actually perform by another person. This is one of the remaining problems which is the cause of many false positive cases in 2019. To alleviate this issue, we use a heuristic method to specify the target faces locations among other characters and a heuristic distance based method. Overall, our team achieves a notable performance in TRECVID INS 2020 by using only pretrained models and heuristic methods.

## 1.2 Introduction

In this paper, we propose a heuristic approach to search for specific person doing specific action. For the person search, the same system as in 2019 is used so that the only difference lies in the action search. For action search, we focus on solving the case when both the target person and the desired action appear in the shot but that action is actually perform by another person. This was one of the remaining problems of TRECVID INS in last year, which caused lots of false positive cases.

Depend on characteristic of action topics, the queries are split into two categories: Human-Object Interaction (HOI) and Facial Expression (FE). HOI is for actions associated with some objects such as: holding phone, sitting on couch,... while crying and laughing are

considered FE actions. For HOI actions, EfficientDet is used for object detection. Moreover, to ensure the the action is conducted by the correct person, we propose to use the heuristic approach based on predicted locations of target faces and the distance between the face and the object. For FE actions, an ensemble model ESR is used. As only facial information is needed to recognize expressions, predicted locations of target faces in top person shots is employed for the same purpose.

Overall, our team achieves a notable performance in TRECVID INS 2020 by using only pretrained models and a heuristic method to ensure the action belong to the target.

## 1.3 OUR APPROACH

### 1.3.1 Person Search

We use the same system for face search as in INS 2019 [1] which includes face detector, descriptor, and matching. For face detection, MTCNN [2] is used, then faces are directly fed into a VGGFace2 [3] model for facial features. For faces in query, an extra step is applied to exclude 'bad' faces which make significant reduction in search performance. To remove such 'bad' faces, we employ a method proposed in [4] which uses SVM to classify if a face is 'good' or 'bad'. Finally, we use cosine similarity for face matching, specifically, we utilize mean-max similarity:

$$sim(query, shot_i) = \frac{1}{N}\sum_{k=1}^{N}(max_{l=1,2,..,M}(cos(desc_k^{query}, desc_l^{shot_i})) \tag{1.1}$$

where $cos(A,B) = \frac{A \cdot B}{\|A\| \cdot \|B\|}$ with $A$ and $B$ are the feature vectors of faces in query and faces in shot, respectively. Here, $N$ is the number of examples in the query set and $M$ is the number of faces in the current shot. The variable $desc_k^{query}$ is the descriptor of the $k$-th face in query while $desc_l^{shot_i}$ is the descriptor of $l$-th face in $i$-th shot.

### 1.3.2 Action Search

For action search, we first split the given queries into two types of actions: Human-Object Interaction (HOI) actions and Facial Expression (FE) actions. Further, to ensure the action is truly performed by the target person, we propose a heuristic method based on the person search result and the distance between the target person and the desired object.

**Table 1.1:** Relevant actions in the MSCOCO and FER+ datasets compared to topics of TRECVID INS 2020

| TRECVID INS 2020 Actions | Related Classes in COCO (Object Detection) | Related Classes in FER+ (Facial Expression Recognition) |
|---|---|---|
| **sit_on_couch** | couch | _ |
| **hold_paper** | book | _ |
| **drinking** | wine_glass, bottle, cup, bowl | _ |
| **crying** | _ | sad |
| **laughing** | _ | happy |
| **holding_phone** | cell_phone | _ |

#### 1.3.2.1 Heuristic target faces identification in high-ranking shots

After the person search stage, we attain shots with a high possibility for having the target person; however, specific faces locations for the target person in those shots are unknown. In this paper, a heuristic approach is used to predict which faces in a top shot belong to the target. First, a person search is carried out using maximum pairwise similarity:

$$sim(query, shot_i) = max_{l=1,2,..,M}(max_{k=1,2,..,N}(cos(desc_k^{query}, desc_l^{shot_i})) \tag{1.2}$$

where $cos(A,B) = \frac{A \cdot B}{\|A\| \cdot \|B\|}$ with $A$ and $B$ are the feature vectors of faces in query and face in shot respectively. Here, $N$ is the number of examples in the query set and $M$ is the number of faces in the current shot. The variable $desc_k^{query}$ is the descriptor of the $k$-th face in query while $desc_l^{shot_i}$ is the descriptor of $l$-th face in $i$-th shot. Next, top-S shots with the highest similarity scores are selected because these shots have a high possibility for containing the target person. A heuristic method is then used to determine which faces belong to the target person. Specifically, we make two assumptions: 1) the target person appears in all frames of a shot; 2) a face with the highest score in each frame is the target face. Using the same idea of maximum pairwise similarity for shot score, a face score is derived as below:

$$sim(query, face) = max_{k=1,2,..,N}(cos(desc_k^{query}, desc_face)) \tag{1.3}$$

In summary, a shot with F frames will have F faces represented for the target person.

#### 1.3.2.2 HOI Action Type

HOI is a special case of Visual Relationship detection (VRD) where we need to detect different types of triplets in an image. Here, a triplet is represented for the two objects and their relationship. For example, a person is watching TV, a book is on a table, two men shake hands,... For the case of HOI, one object must be a human and the other must be an object that person is using. Moreover, their relationship must be an action but not their position. In this paper, a HOI is defined as a triplet of a subject, an object and their relationship which must be an action.

For addressing the HOI problem, different approaches have been proposed. But the most common way is to first use an object detection model, then all detected objects are fed into a HOI detection model. Basically, a detector is used to first detect all objects in an image, then all human-object pairs generated from those objects are considered proposals for VRs. For two objects in each pair, their bounding boxes and the object classes will be used to decide the type of their relationship. Specifically, in VRD, it is common to use the relative positions of two bounding boxes and/or their visual information, i.e., the object type, the way the person interacts with that object, and the around context. But one disadvantage of these approaches is that they are very expensive to apply for a large-scale dataset like BBC EastEnders. Additionally, the performance of current approaches is quite low on typical benchmark like HICO-DET (PPDM [5] only achieves 24.5% mAP on HICO-DET), which makes it hard to be employed in practice.

For the above reasons, we do not use VRD models in this work but propose a heuristic method based on the distance between target face and desired object. As mentioned in 1.3.2.1, a heuristic method is employed to clearly specify faces locations of the target (in frames) using the top-ranking shots. For objects locations, we use EfficientDet detector [6]. Finally, a distance between the human face bounding box ($human$) and the object bounding box ($object$) is calculated as follows:

$$d(human, object) = \frac{d(cent_{human}, cent_{object})}{diag_{enclosing}} \tag{1.4}$$

where $enclosing$ is the smallest enclosing box of the human face box and the object box. We assume that the smaller the distance is, the more chance the two boxes will form a positive relationship. Apparently, the face is used instead of person bounding box to ensure the action is performed by the correct person.

Based on the calculated distances, a search is carried out to find positive action shots. Each shot is represented by an action score which can be computed as follows:

**Algorithm 1:** How to compute HOI action score for one shot for one topic

```
Result: HOI Action Score
ListOfDist := [];
for each frame fr in shot do
    Choose the face f with highest score in fr as the target face; for each detected
     object o in fr do
        // Based on Table 1.1
        if the object o is related to the action of the mentioned topic then
            Compute d(f, o);
            Append to ListOfDist;
        end
    end
end
return min(ListOfDist);
```

Overall, this algorithm is used to compute HOI action score for all shots in the database. Then, all the computed scores will be used to rerank the shots for action search. Notice that only objects with confidence score larger than 0.5 are chosen as candidates.

#### 1.3.2.3 FE Action Type

For facial expression actions, we adopt a similar approach based on the person search result but without the need of calculating the distance. Unlike HOI actions, FE actions can be recognized based on only facial information. Therefore, using only result from the person search stage is enough to ensure the facial expression is performed by the correct person. In summary, we use a deep facial expression model together with a heuristic person identification (as denoted in 1.3.2.1) to search for positive shots. In this work, an efficient ensemble method called ESR [7] is employed for facial recognition. The ESR is applied only on the target faces which are the outputs of the heuristic face identification 1.3.2.1. Given a face, this ensemble model will predict 10 facial expressions. If a shot has $F$ frames and each of which is predicted to contain one target face, that shot will have $10 * F$ facial expressions in total. For simplicity, we scoring the shot by using the frequency of the topic-related expressions:

**Algorithm 2:** How to compute FE action score for one shot for one topic

```
Result: FE Action Score
core_exp := 0;
for each frame fr in shot do
    Choose the face F with highest score in frame fr as the target face;
    F is fed into an ESR model for a list of facial expressions ListOfFacialExp;
    for each fe in ListOfFacialExp do
        // Based on Table 1.1
        if the expression fe is related to the action of the mentioned topic then
            core_exp += 1
        end
    end
end
return core_exp/len(ListOfFacialExp);
```

After FE scores are calculated for all shots, they will be used to rerank the shots for action search. More details will be discussed in the next section 1.3.3.

### 1.3.3 Fusion

To fuse the person search and action search, we select top shots from person search rank list and rerank those shots based on action score. Basically, top $S$ shots are selected from the person search stage, those shots are considered to have the highest possibility of carrying the target person. Because all $S$ shots are assumed to have the target, now we just have to select shots with the highest action scores. Finally, after being filtered by both person and action score, top-1000 of remaining shots are returned as the output of our system.

## 1.4 EVALUATION

We use four different fusion settings for submissions. Each setting differs only in the variable $S$ - the numbers of top shots selected from the person result before conducting action search. Table 1.2 shows all the official evaluation results of our submissions.

## 1.5 CONCLUSION

Our teams achives notable performance in TRECVID INS 2020 using only pretrained models and proposed heuristic methods. Specifically, we propose an effective heuristic approach

**Table 1.2:** Result of our submitted 4 runs on Instance Search task of TRECVID 2020.

| Run | mAP | Top S shots from person search |
|---|---|---|
| **NII_UIT.20.1** | 0.088 | 15K |
| **NII_UIT.20.2** | 0.091 | 10K |
| **NII_UIT.20.3** | 0.091 | 10K |
| **NII_UIT.20.4** | 0.087 | 15K |

to ensure the action is performed by the target person. Last year, this was one of the major problems which caused many false positive cases. Our proposed heuristic method has partially solved this problem; however, there are many given assumptions which should be discussed more in future work.

# Chapter 2

# Video Summarization (VSUM)

## 2.1 Abstract

Video summarization is a task to produce a short video skim that preserve the most important content of the original video. To promote research in semantic video summarization, this task has added in TRECVID 2020 where each team have to build a system that summaries major life events of specific characters. In this work, we proposed a framework to generate a final summary by combining the score of person face and a self-attention based network. Our method has experimented on the BBC Eastenders dataset.

## 2.2 Introduction

Summarizing the video is to reduce the size and keep the amount of high value information by selecting and presenting the most informative or interesting materials for potential users. Output of the summary usually is a small number of keyframes or shoter video sequence composed of keyshots. Video Summarization (VSUM) is a new task in TRECVID 2020 where each team have to build a system that summaries major life events of specific characters. The main task is to summarize the major life events of each character in the BBC Eastenders TV series within the specified time frame of the series. Some example of major events are more likely to be: the birth of a child, a divorce, the passing of a loved, etc. Specifically, we have to build a framework to generate summaries for three specified characters of the series, where each character will consist of a set of 4 example frame images, the time frame of the series (Start Shot # and End Shot #), and the maximum length and number of shots for each run.

The most recent approaches [8, 9, 10] for the video summarization task include three


**Figure 2.1:** The phases in our summary framework

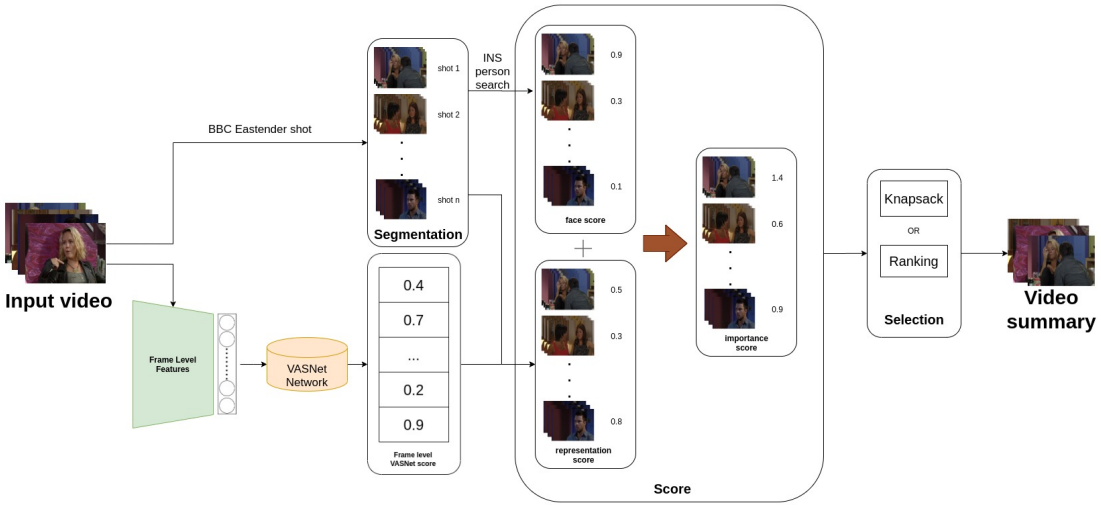


**Figure 2.2:** Our summary framework for TRECVID VSUM task 2020

phases: i) Splitting video into shots. ii) Scoring each shot based on some specific criteria. iii) Selecting the appropriate shots to generate the final summary. Therefore, in this work, we decided to divide our framework into three modules separately, as Figure 2.2: segmentation, score, and selection. The first module is segmentation, where input video will be split into short segments by using available shot-time information of the BBC EastEnders dataset. Then, based on the person face and representation score, the score module will predict importance score of each shot. Finally, for selection, keyshots are then selected with the Knapsack algorithm and then concatenated to produce the final video summary.

## 2.3 Our Approach

### 2.3.1 Segmentation

Videos are divided into short segments and the desired summary is generated by finding a subset of segments that maximizes the total importance score in the summary. In our approach , we use the shot-time information that is provided in BBC EastEnders dataset to split input videos into shots as segment.

### 2.3.2 Score

To summarize videos by desired person and capture the major life events of them, we calculated importance score of each shot by combining the person face and representation score respectively.

#### 2.3.2.1 person face score

For a person score, a similarity score from the INS person search system is chosen. Concretely, the face of a character is used to consider whether the person should be a target. Here, we use the same system for face search as in INS 2019 [1] which includes face detector, descriptor, and matching. For face detection, MTCNN [2] is used, then faces are directly fed into a VGGFace2 [3] model for facial features. For faces in query, an extra step is applied to exclude 'bad' faces which make significant reduction in search performance. To remove such 'bad' faces, we employ a method proposed in [4] which uses SVM to classify if a face is 'good' or 'bad'. Finally, we use cosine similarity for face matching, specifically, we utilize mean-max similarity:

$$sim(query, shot_i) = \frac{1}{N}\sum_{k=1}^{N}(max_{j=1,2,..,M}(cos(desc_k^{query}, desc_j^{shot_i})) \tag{2.1}$$

where $cos(A,B) = \frac{A \cdot B}{\|A\| \cdot \|B\|}$ with $A$ and $B$ are the feature vectors of faces in query and faces in shot, respectively. Here, $N$ is the number of examples in the query set and $M$ is the number of faces in the current shot. The variable $desc_k^{query}$ is the descriptor of the $k$-th face in query while $desc_j^{shot_i}$ is the descriptor of $j$-th face in $i$-th shot.

#### 2.3.2.2 representation score

In [11] addressed that a summarization algorithm relying on visual content in the video only without considering the high-level semantic information will erroneously eliminate important frames. Currently, there are many methods for video summarization using bi-directional LSTM[12], GRU[13] and soft attention[14] to capture semantic information and achieved high performance on two famous datasets, TvSum[8] and SumMe[9]. A proposal for video keyshots summarization, VASNet[15] that is a approach to sequence to sequence transformation for video summarization based on soft, self-attention mechanism , is demonstrated its performance on these datasets and is currently state of the art method. Thus, we using this method to compute the representation score of each shot.

For each video, we used GoogLeNet network[16] trained on ImageNet[17] to extract features $X = (x_0, x_1, ..., x_n)$, $n$ is a total of frames in the video from the pool5 layer of every frames . The VASNet model takes an input sequence $X$ and calculates importance score of each frame $Y = y_0, y_1, ..., y_n, y_i = [0, 1)$ - score of frame $i^{th}$. After that, we segment the video into shots following available shot-time BBC EastEnders datatset. For each detected shot $s \in L$, we calculate score $s_i$ as follow:

$$s_i = \frac{1}{l_i} \sum_{x=1}^{l_i} y_i, x \tag{2.2}$$

where $y_i, x$ is score of $x^{th}$ frame within shot $i$ and $l_i$ is a total of frames in $i^{th}$ shot.

#### 2.3.2.3 importance score

The importance score of each shot in a video is calculated by the sum of person face and representation score.

$$importance\_score = person\_score + representation\_score \tag{2.3}$$

Furthermore, Yale Song et al. [8] empirically found that a shot length of two seconds is appropriate for capturing the context with good visual coherence. Thus, shots less than 2 seconds in length will have importance score of 0. In addition, we also observed that there are some redundant shots at the beginning and the end of a video for advertising or introduction. Therefore, we have assigned 0 score manually for such shots.

### 2.3.3 Selection

In TRECVID VSUM 2020 task, each team have to submit 4 runs for each character following the Figure 2.3.

The last phase after having the importance score of shots is selecting a subset of shots that maximizes the total importance score in the summary. To select keyshots for the summary, we first chose 1000 shots with the greatest person score, and then ranked them in order of descending representative score, subset of shots $K$ as a result. For run 1 and 3, we picked top 5 and 15 shots in $K$ respectively, and concatenated them into a final video summarization. On the other side, for run 2 and 4, keyshots are selected with Knapsack algorithm according to [8] by length of shots and importance score of shots.

| Character | Janine | Ryan | Stacey |
|---|---|---|---|
| Start Shot # | shot175_1 | shot175_1 | shot175_1 |
| End Shot # | shot185_1736 | shot185_1736 | shot185_1736 |
| Images | Images | Images | Images |
| Max # Shots Run 1 | 5 | 5 | 5 |
| Max Summary Length Run 1 | 150 seconds | 150 seconds | 150 seconds |
| Max # Shots Run 2 | 10 | 10 | 10 |
| Max Summary Length Run 2 | 300 seconds | 300 seconds | 300 seconds |
| Max # Shots Run 3 | 15 | 15 | 15 |
| Max Summary Length Run 3 | 450 seconds | 450 seconds | 450 seconds |
| Max # Shots Run 4 | 20 | 20 | 20 |
| Max Summary Length Run 4 | 600 seconds | 600 seconds | 600 seconds |

**Figure 2.3:** Specifies of VSUM task 2020

**Table 2.1:** Result of our submitted 4 runs on Video Summarization task of TRECVID 2020.

| Character Name | Run Number | #Shots | Summary Length |
|---|---|---|---|
| Janine | 1 | 5 | 16.5 |
| | 2 | 10 | 21 |
| | 3 | 15 | 67.9 |
| | 4 | 20 | 42 |
| Ryan | 1 | 5 | 12.2 |
| | 2 | 10 | 21 |
| | 3 | 15 | 43.1 |
| | 4 | 20 | 42.1 |
| Stacey | 1 | 5 | 16.7 |
| | 2 | 10 | 20 |
| | 3 | 15 | 49 |
| | 4 | 20 | 42 |

## 2.4 EXPERIMENTS

### 2.4.1 Dataset

BBC EastEnders dataset includes 244 video files, approximately 464h in MPEG-4 format. Also, the data comprises transcripts, and a small amount of additional metadata. In the task this year, we have to generate video summary about the major event life of each character within from video175 to video185.

### 2.4.2 Result

We have created 4 runs for this year's three characters on the BBC EastEnders dataset. The detail information of video summaries are described in table 2.1

## 2.5 CONCLUSION

By participating in the Video Summarization task in TRECVID 2020, we proposed a approach to generate the video summary for this task. Our framework using the face similarity score and VASNet score was performed on BBC dataset.

# Chapter 3

# Disaster Scene Description and Indexing (DSDI)

NII_UIT team participated in DSDI task in "loose" collaboration with NIICT (NICT and NII) team and VAS (Hitachi) team. We shared ideas and findings through roughly monthly meetings, developed systems independently, and exchanged prediction results, namely, frame-level and clip-level scores for fusion runs.

NII_UIT DSDI runs are bacisally composed of fusion runs of NIICT and VAS runs. Detailed explanation of NIICT and VAS runs can be found in respective notebook papers [18, 19]. The description of NII_UIT DSDI runs is as follows:

| Run | mAP | Systems | Description |
|---|---|---|---|
| NII_UIT.nii1 | 0.374 | **VAS 3**<br>**NICT-3104** | A clip-level fusion is performed on clip-level scores<br>with balance parameters 5/6 (**VAS 3**) and 1/6 (**NICT-3104**). |
| NII_UIT.nii2 | 0.248 | **VAS-D13** | A single model trained with an SE-IBN-Resnet-50 under<br>a multi-label smoothing and class imbalance cost function (Focal-loss). |
| NII_UIT.nii3 | 0.331 | **NICT-3120**<br>**C03**<br>**D13** | A frame-level fusion is performed on the geometric mean of frame-level scores.<br>The frame-level score is obtained for each commonly selected frame.<br>The final clip-level score is the maximum frame-level score in each clip. |

Note that we have five systems: 1) NIICT (NICT and NII) system **NICT-3120**, 2) NIICT (NICT and NII) system **NICT-3104**, 3) VAS (Hitachi) system **C03**, 4) VAS (Hitachi) system **D13**, and 5) VAS (Hitachi) system **VAS 3**.

# Bibliography


[1] M. Klinkigt, D. Le, A. Hiroike, H. Q. Vo, M. Chabra, V. Dang, Q. Kong, V. Nguyen, T. Murakami, T. Do, T. Yoshinaga, D. Nguyen, S. Saptarshi, T. D. Ngo, C. Limasanches, T. Agrawal, J. Vora, M. Ravikiran, Z. Wang, and S. Satoh, "NII hitachi UIT at TRECVID 2019," in *2019 TREC Video Retrieval Evaluation, TRECVID 2019, Gaithersburg, MD, USA, November 12-13, 2019*, G. Awad, A. A. Butt, K. Curtis, Y. Lee, J. G. Fiscus, A. Godil, A. Delgado, J. Zhang, E. Godard, L. L. Diduch, A. F. Smeaton, Y. Graham, W. Kraaij, and G. Quénot, Eds. National Institute of Standards and Technology (NIST), 2019. [Online]. Available: https://www-nlpir.nist.gov/projects/tvpubs/tv19.papers/nii_hitachi_uit.pdf

[2] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[3] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 2018, pp. 67–74.

[4] H.-Q. Vo, V.-M.-H. Dang, V.-T. Nguyen, and D.-D. Le, "Noise removal based query preprocessing to improve face search performance in large scale video databases," in *Proceedings of the Tenth International Symposium on Information and Communication Technology*, 2019, pp. 357–361.

[5] Y. Liao, S. Liu, F. Wang, Y. Chen, C. Qian, and J. Feng, "Ppdm: Parallel point detection and matching for real-time human-object interaction detection," in *CVPR*, 2020.

[6] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 781–10 790.

[7] H. Siqueira, S. Magg, and S. Wermter, "Efficient facial feature learning with wide ensemble-based convolutional neural networks," pp. 1–1, Feb 2020. [Online]. Available: https://www2.informatik.uni-hamburg.de/wtm/publications/2020/SMW20/SMW20.pdf

[8] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, "Tvsum: Summarizing web videos using titles," pp. 5179–5187, 2015.

[9] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating summaries from user videos," pp. 505–520, 2014.

[10] M. Otani, Y. Nakashima, E. Rahtu, and J. Heikkila, "Rethinking the evaluation of video summaries," pp. 7596–7604, 2019.

[11] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Video summarization with long short-term memory," in *European conference on computer vision*. Springer, 2016, pp. 766–782.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, p. 1735–1780, Nov. 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[13] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[14] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[15] J. Fajtl, H. S. Sokeh, V. Argyriou, D. Monekosso, and P. Remagnino, "Summarizing videos with attention," 2018.

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[17] 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*, vol. 115, no. 3, pp. 211–252, 2015.

[18] S. Iwasawa, K. Uchimoto, Y. Kidawara, and S. Satoh, "Is automl a practical way of tackling dsdi task?" in *TRECVID Notebook Paper*, 2020.

[19] S. Okazaki, Q. Kong, M. Klinkigt, and T. Yoshinaga, “Hitachi at trecvid dsdi 2020,” in *TRECVID Notebook Paper*, 2020.