DOI: 10.3217/978-3-85125-869-1-10

SliTraNet: Automatic Detection of Slide Transitions in Lecture Videos using Convolutional Neural Networks

Aline Sindel[1], Abner Hernandez[1], Seung Hee Yang[2], Vincent Christlein[1] and Andreas Maier[1]

Abstract— **With the increasing number of online learning material in the web, search for specific content in lecture videos can be time consuming. Therefore, automatic slide extraction from the lecture videos can be helpful to give a brief overview of the main content and to support the students in their studies. For this task, we propose a deep learning method to detect slide transitions in lectures videos. We first process each frame of the video by a heuristic-based approach using a 2-D convolutional neural network to predict transition candidates. Then, we increase the complexity by employing two 3-D convolutional neural networks to refine the transition candidates. Evaluation results demonstrate the effectiveness of our method in finding slide transitions.**

I. INTRODUCTION

Nowadays, there is a huge number of online learning material available to students and researchers. Lecture videos uploaded by the universities to video sharing platforms such as YouTube or to in-build video platforms are accessible from anywhere and at any time. The high amount of video material makes it tedious for the user to search for specific content by browsing through the individual videos. Hence, video summarization can help to quickly grasp the overview of the lecture video. This can be done by the automatic detection of slide transitions to extract the slide and time stamp at each slide change. Automatic detection of slide transitions can also support the lecturer in creating lecture notes. In combination with the audio transcript of the lecture video, the extracted slides can be automatically inserted into the audio text based on their time stamp. For instance, the free video-to-blog post conversion software AutoBlog [21] automatically extracts the transcript of a lecture video to generate a blog post [9]. So far, the slides are manually inserted into the blog text. However, using our slide transition detection method, the software could be extended.

The variety in the types of lecture videos makes the task challenging. For example, the lecture slides can be full screen with the lecturer screen inserted as a small window on top, or the lecture slides can be depicted next to the view of the lecturer. Further, memes (e.g. animations and short videos) to illustrate the lecture content can be inserted into the lecture video. Memes and the actual slides can have very similar frames from the style and color distribution. Thus, lecture videos that not only contain the slides and the speaker's view, but also these meme videos make the task even more difficult.

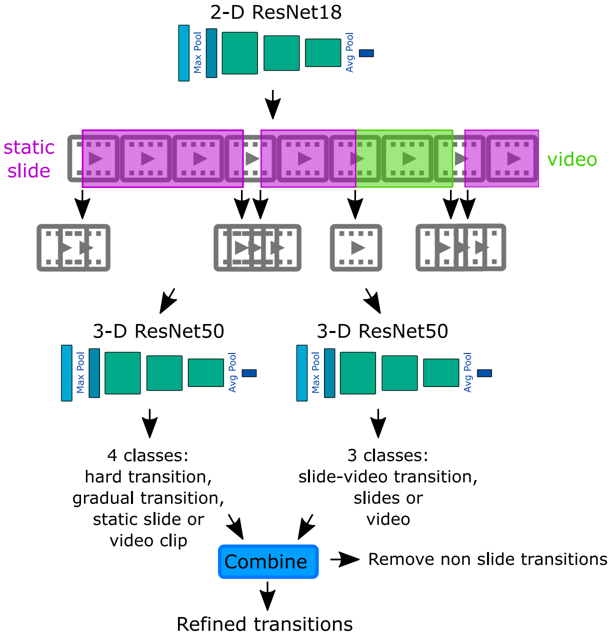

2-D ResNet18
Max Pool
Avg Pool
static slide
video
3-D ResNet50
Max Pool
Avg Pool
3-D ResNet50
Max Pool
Avg Pool
4 classes: hard transition, gradual transition, static slide or video clip
3 classes: slide-video transition, slides or video
Combine
Remove non slide transitions
Refined transitions

Fig. 1: Overview of our SliTraNet for slide transition detection: First, we predict initial slide-slide or slide-video transition candidates by comparing each frame (cropped to the slide content) to its respective anchor frame using a 2-D ResNet. At the transition candidate positions, we extract overlapping video clips with a length of eight frames from the cropped video and the raw video. Two 3-D ResNets have been trained to extract spatio-temporal features to classify the cropped video clips into hard or gradual transitions, static slides or video sequences and the raw video clips into slide-video transitions, slide sequences or video sequences. Lastly, we combine the class predictions of both 3-D ResNets to exclude transitions mutually classified as video sequence.

In this paper, we propose a deep learning method for the detection of slide transitions in lecture videos, which we train and test on a dataset that contains video sequences of lectures with slides, speaker views, and memes. To detect the slide transitions, we present a multi-step approach. First, we predict initial transition candidates by inserting a 2-D convolutional neural network (CNN) into a heuristic-based approach. Then, we extract spatio-temporal features at the candidate positions using two 3-D CNNs to exclude transitions that were classified as video sequences.

This work was supported by the project "MEOW" funded by "DAAD IP Digital"

[1]Pattern Recognition Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 91058 Erlangen, Germany {aline.sindel, abner.hernandez, vincent.christlein, andreas.maier}@fau.de

[2]Department of Artificial Intelligence in Biomedical Engineering, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 91058 Erlangen, Germany seung.hee.yang@fau.de

II. Related Work

This section summarizes related works in the field of slide transition detection, scene boundary detection, and video thumbnail selection.

A. Slide Transition Detection

Traditional approaches to slide detection focus on low-level features to measure the similarity across adjacent frames. For example, the maximum peak of the color histogram and difference in entropy for horizontal lines were used to detect slide changes in [14]. Often the use of histograms for slide detection is supplemented with other algorithms to detect features such as faces, or text [20], [29]. Similarly in [2], histograms are utilized for shot boundary detection as part of a larger scheme involving shot classification, slide region detection, and slide transition detection.

The variance in image scaling and rotation can be handled by the Scale Invariant feature transform (SIFT) algorithm. This approach detects slide transitions when the SIFT similarity is under a defined threshold. Features extracted using the SIFT algorithm have shown good slide detection accuracy rates in [10], [22] and with slide alignment [28]. SIFT features can also be used with sparse time-varying graphs [17], where the graph models slide transitions. The temporal modeling of slide transitions can also be conducted using a Hidden Markov Model (HMM), where the states of the model correspond to an individual slide [5], [24], [3], [4]. The likelihood of the states are computed with a correlation measure and the most probable sequence of slides is calculated using the Viterbi algorithm.

The current study approaches the slide transition detection problem by using 3-D CNNs which can learn spatio-temporal features that are useful for detecting slide transitions. However, the training time and memory consumption can be problematic. Therefore, Residual Networks (ResNet [8]) have been suggested by [18] for this task. They propose a novel residual block that contains an extra 1×1 3-D convolutional layer to the shortcut connection layer. They show better results for ResNet compared to the traditional slide transition approaches on their to 6 frames per second temporally downsampled dataset. In [6], a Dual Path Network (DPN) [1] that combines both ResNeXt and DenseNet is proposed. Further, they introduce a Convolutional Block Attention module to their network that sequentially infers a 1-D channel attention map, followed by a 2-D spatial attention map, and lastly a 1-D time attention map. Further improvements in the F_1-score were obtained compared to traditional approaches or with ResNets alone.

B. Scene Boundary Detection

A related field of work is scene boundary detection or shot boundary detection (SBD) [11]. Traditionally, SBD relied on the same low-level features such as histograms. However, the issue of detecting changes is complex and requires attention to the variability of transitions. Detecting gradual transitions is a particularly difficult problem and recent studies on SBD now take into consideration the presence of sharp cut transitions and gradual transitions. For example, a 3-D CNN-based model from [7] was combined with an SVM classifier to label frames as being either normal, a gradual transition, or a sharp transition. In [15], both types of transitions are detected by separate 3-D CNNs. A similar approach using deep CNNs was taken by [27] where SBD was implemented via a three stage process; candidate detection, cut transition detection, and gradual transition detection. TransNet [26] and TransNet2 [25] use Dilated DCNNs to detect sharp and gradual transitions.

C. Video Thumbnail Selection

Another related area is video thumbnail selection, which summarizes the video content by selecting a representative frame as the thumbnail. To extract the representative frames, learning-based approaches have been proposed that take the user's perspective selection of representative frames into account [12], [19]. Based on visual features, the videos are classified according to image quality, visual details, user attention, and display duration [12], or different types of camera motion [19]. Approaches also combine the visual content with side semantic information such as the title or transcript for query-dependent thumbnail selection [16] or to visually enrich the thumbnail with keywords [30].

III. Methodology

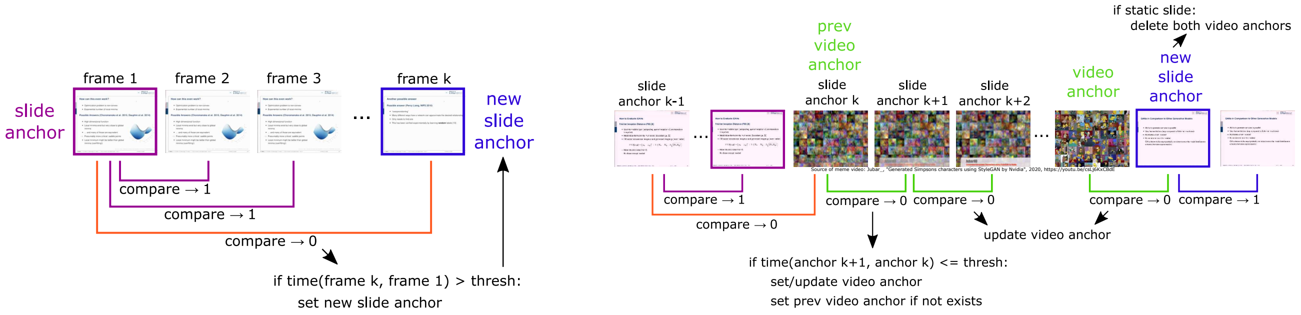
In this section, our method for slide transition detection is presented. We describe the network architectures and introduce training and inference of the different parts of the pipeline.

A. Overview of SliTraNet

SliTraNet is composed of three convolutional neural networks, which are all separately trained for the three different tasks and combined for inference, see Fig. 1. We process the complete data once by applying a 2-D ResNet18 [8] to pairs of each frame with its anchor frame resulting in initial slide-slide or slide-video transition candidates. For the refinement step, we increase the complexity of the networks by using two 3-D ResNet50s and apply these to video clips extracted from the transition candidate positions. A short video clip of eight frames can contain a sequence with one hard transition, a gradual transition, a static sequence of the same slide or a sequence of video frames, such as a short animation, a speaker view, or a meme. We train one 3-D ResNet for these four classes and another 3-D ResNet to distinguish slide-video transitions, slide sequences, and video sequences. Based on the class predictions, we exclude transition candidates that were classified as video sequence by both networks.

B. Initial Transition Candidates Estimation

We train a 2-D ResNet18 for the discrimination task whether two images are from the same slide (class 1) or not (class 0). For this task, we concatenate both images along the color channel dimension to obtain 6 channels for RGB input or 2 channels for grayscale input and modify the



Source of meme video: Jubar_, "Generated Simpsons characters using StyleGAN by Nvidia", 2020, https://youtu.be/csLj6KxCBdE

(a) Detection of a static slide sequence

(b) Detection of a video frame sequence

Fig. 2: Comparison to anchor frame using a neural network to detect static slide sequences and video sequences.

input channels of the ResNet18 [8] architecture accordingly. For training, we generate the same number of positive and negative pairs. For the negative pairs, we first select frames from the neighboring slides for each slide and then fill the rest with randomly chosen frames that have a different slide id. For the optimization, we employ the binary cross-entropy loss.

To predict the transition candidates, we plug the neural network into a heuristic-based approach, as illustrated in Fig. 2a and 2b. We compare each frame to an anchor frame by the neural network to search for static slides (Fig. 2a). As long as both frames are classified as the same, we keep the anchor and as soon as the two frames are classified as different, we set the anchor to the current frame. A static slide is detected if the time, measured in number of frames, is higher than a threshold. This general idea is borrowed from Perelman [23], which uses the absolute difference of the blurred grayscale versions of the frames. Since the lecture videos also contain video sequences without slides, we extended the approach further by adding two video anchors, see Fig. 2b. If a frame difference is detected by the neural network and the time from the current frame k to the anchor $k-1$ is smaller or equal to the threshold, the video anchor and previous video anchor are set to the current frame k. As long as the frames are not classified as the same, the video anchor is updated. After the next static slide sequence is detected, a video sequence is recorded from the previous video anchor to the video anchor and both anchors are deleted. The slide-slide and slide-video transition candidates are determined from the detected static slide and video sequences.

C. Transition Candidates Refinement

Since the video sequences can also contain static frames that might be classified as static slides using the deep-heuristic-based approach, a refinement step is necessary to reduce the number of false positives. To better exploit the spatio-temporal character of the video, we train a 3-D ResNet50 using cross-entropy loss for the multi-task classification problem that assigns a short video clip of eight frames to one of the classes: hard transition, gradual transition, static slide and video. The network architecture is depicted in Fig. 3, which is slightly adapted from the 3-D ResNet backbone in [13]. For the initial layers and 3-D max pooling, we modified the strides of the temporal dimension to be 1 to only reduce the spatial dimensions.

The slides of lecture videos are not necessarily filling the full screen, but can be placed on top of some background. In our particular lecture video dataset, the memes, animations and speaker video sequences are full screen in contrast to the slides, see Fig. 4. Using this knowledge, we use the raw video input to train our second 3-D ResNet50 to classify the short clip into slide-video transitions, slide sequences or video sequences.

For training both networks, we extract video clips at striking positions such as placing the middle of the clip (plus minus one frame) at the position of the hard transition, the begin, middle and end of the gradual transition and in the middle of a static slide sequence or at some equally spaced positions within the video sequence. For the second task, the slide-video transitions occur only rarely in the dataset in comparison to slide sequences or video sequences. Hence, we use the weighted cross-entropy loss to account for the frequency of the classes.

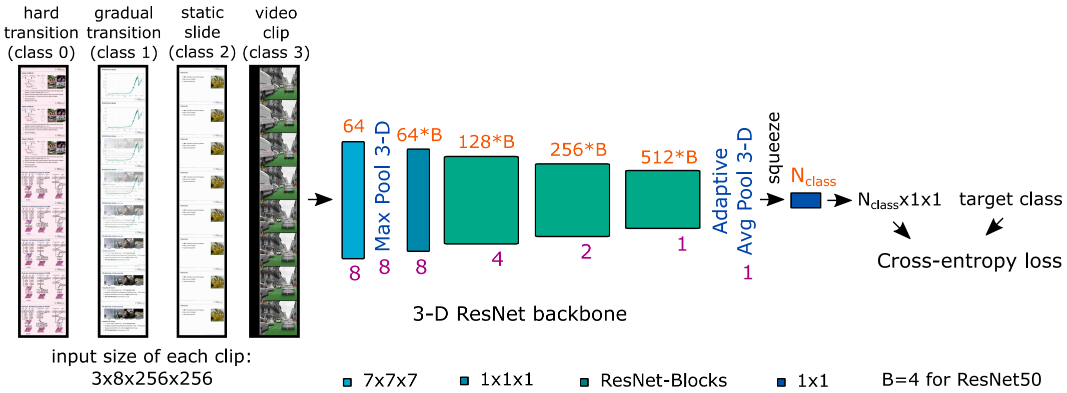
During inference, we use the predictions of the deep-heuristic-based approach to extract the video clips to feed them to the 3-D ResNets and based on the output of both networks, we filter out slide transition candidates that were classified to be video sequences by both networks.

IV. EXPERIMENTS AND RESULTS

In this section, we describe our dataset and measure the performance of our method.

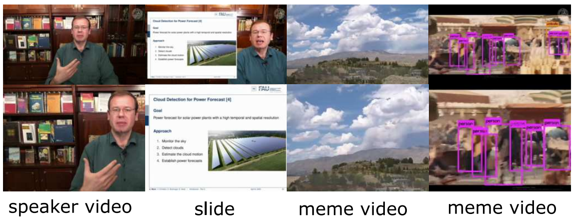
A. Lecture Video Dataset

The dataset comprises a subset of lecture videos from two courses in the field of deep learning and medical image processing of the Pattern Recognition Lab, FAU Erlangen-Nuremberg. The videos are recorded in Full HD with 25 frames per second and range between a duration of 6 to 33 min. The slides of one course are in the format 4 : 3 and of the other in 16 : 9. The dataset is split into 12 videos for training, 4 for validation and 14 for testing. To feed the data to the network, the frames are cropped to the content of the slides except for the video-slide differentiation task



Source of meme video (class 3): Mobileye an Intel Company, "Sensing the Driving Scene", 2017, https://youtu.be/42rmGs0Rvtw

Fig. 3: Training of 3-D ResNet50 for the multi-class classification task: detection of hard transitions, gradual transitions, static slides, and videos. For each input clip one class is predicted. The numbers in orange indicate the number of output feature maps of each convolutional layer or block and the numbers in purple denote the output dimension of the temporal dimension.



Source of meme 1: CTP 1111 "Time Lapse of Flammagenitus or Pyrocumulonimbus clouds from the Loyalton Fire", 2020, https://youtu.be/zISr9AA2oow
Source of meme 2: YOLO Object Detection, "YOLO v2", 2016, https://youtu.be/VOC3huqHrss

Fig. 4: Frames of the lecture video dataset. Top row: raw video frames, bottom row: cropped frames.

(see Fig. 4) and for all tasks are scaled to a maximum length of 256 and filled up with zero padding to a patch size of 256×256. The ground truth slide transitions were obtained semi-automatically. Based on the difference of the frames, static slides were roughly detected and were manually corrected at frame level and split into hard and gradual transitions.

B. Implementation Details

We trained all networks from scratch for 100 epochs with early stopping using the following training parameters: learning rate $\eta = 2 \cdot 10^{-4}$, linear decay to 0 starting at epoch 50 for 2-D ResNet18 and 60 for 3-D ResNet50, Adam solver, momentum $(0.9, 0.999)$, batch size of 64 for 2-D ResNet18 and of 32 for 3-D ResNet50 for training and validation and online data augmentation for the training data split (color jittering, horizontal flipping, color inversion, Gaussian blurring with kernel size in range 1 to 21, reversed ordering of the clips and one frame offsets at clip extraction). For inference, the threshold for static slides is set to 8 frames.

C. Qualitative and Quantitative Evaluation

We evaluate our method using precision, recall, and F_1 score of slide transitions for our test dataset. Since the gradual transitions are annotated and predicted by our method as frame intervals, we compare the closest euclidean distances of the start and end points of the predicted and labeled transitions to a threshold of 20. This comparison is performed bi-directionally and the mutually valid counts determine the number of true positive transitions.

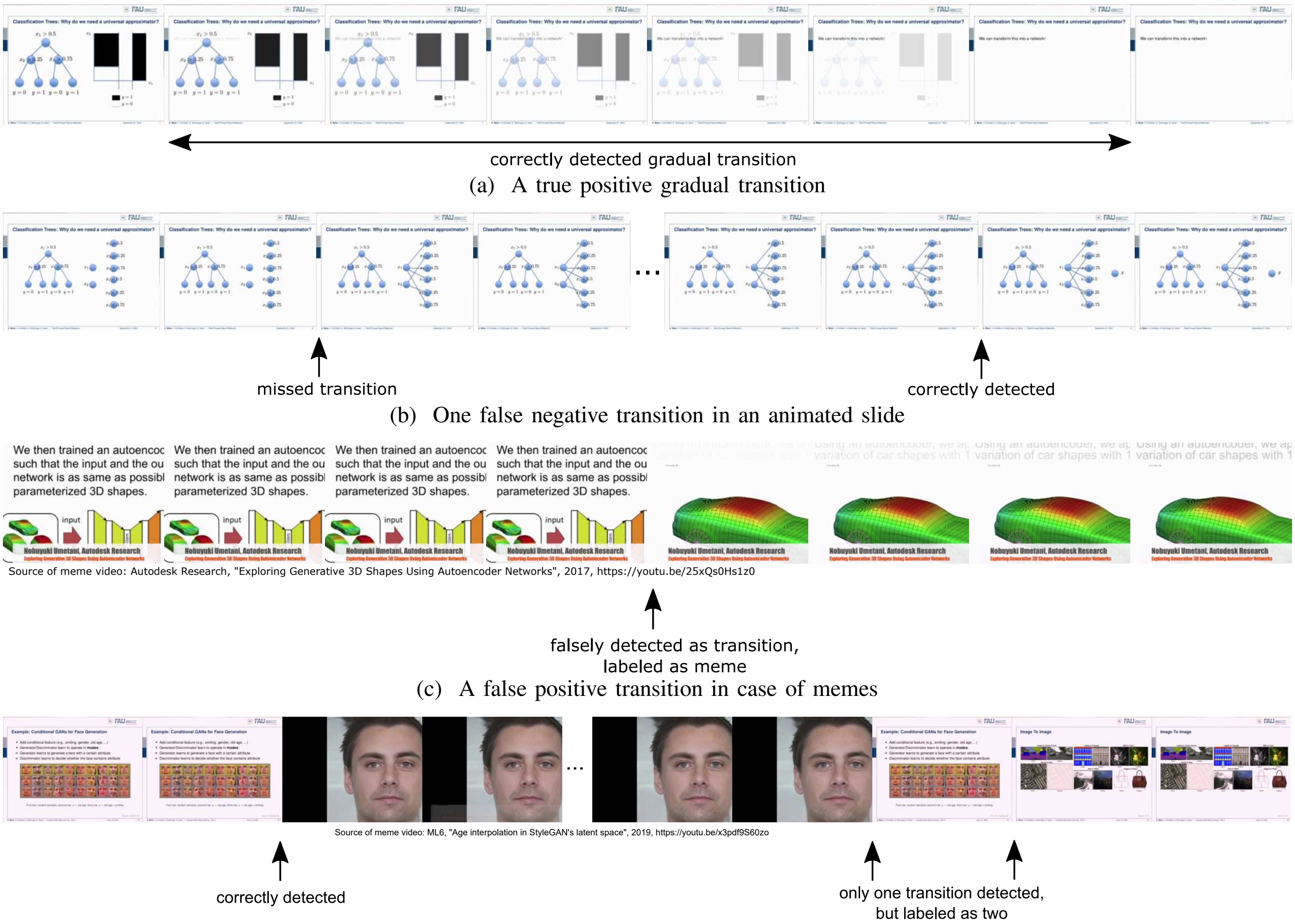
The quantitative evaluation results are summarized in Tab. I. In the top rows, we compare the first step of our approach using the 2-D ResNet18 (trained and tested in RGB and grayscale) to the traditional approach inspired by [23] of using the frame difference with Gaussian blur (kernel size $k_s = (21, 21)$) in RGB and grayscale. From these methods, the grayscale 2-D ResNet achieves the highest F_1 score, which is slightly above 50 %. The 2-D methods have a high recall but a low precision due to their high number of false positives. The frame to anchor comparison detects many false positive transitions for video frames, where short static sequences alternate with motions.

Hence, the second part of our pipeline is necessary to reduce these false positives, whose results are shown in the bottom rows of Tab. I. Using the combination of the first step and the 3-D ResNets a performance gain in the F_1 score of up to 35 % is achieved, i.e., our SliTraNet reaches an F_1 score of almost 90 %, which is closely followed by the combination of difference + 3-D ResNets. This second step maintains the high recognition rate while decreasing the number of false positives, resulting in higher precision, which is partly due to the spatio-temporal convolutions in the 3-D ResNets that recognize the different transition types better than the 2-D approach.

Additionally, we evaluate how the order of the networks influences the result by reversing the order. First, we apply the 3-D ResNet to classify overlapping video clips of length 8 into slide-video, slides and videos. We use the slide-video and slide candidates to apply the next 3-D ResNet to classify the remaining clips into the transition types, static slides and videos. We iterate through the potential transition regions and apply the 2-D ResNet pairwise to localize slide changes. This

TABLE I: Evaluation of precision, recall, and F_1 score of slide transition detection for the test set with 14 videos. In the top rows is the comparison of the first step of the approach: 2-D ResNet18 versus difference with Gaussian blur in both color and grayscale. In the bottom rows the combination of the above methods with the 3-D ResNet50 (in color) and the application of the three networks in reverse order (first 3-D then 2-D) is shown.

	Number of transitions	TP	FP	FN	**Precision**	**Recall**	**F_1 score**
Ground Truth	380	380	0	0	100.00	100.00	100.00
Diff-RGB-blur	992	365	627	15	36.79	**96.05**	53.21
Diff-gray-blur	1011	365	646	15	36.10	**96.05**	52.48
2-D ResNet18-RGB	1188	358	830	22	30.13	94.21	45.66
2-D ResNet18-gray	911	355	556	25	**38.97**	93.42	**55.00**
ResNets-Reverse-RGB-gray	366	303	63	77	82.79	79.74	81.23
Diff-RGB-blur + 3-D ResNet50-RGB	435	364	71	16	83.68	**95.79**	89.33
Diff-gray-blur + 3-D ResNet50-RGB	442	364	78	16	82.35	**95.79**	88.56
SliTraNet-RGB-RGB	453	357	96	23	78.81	93.95	85.71
SliTraNet-gray-RGB	408	354	54	26	**86.76**	93.16	**89.85**


correctly detected gradual transition
missed transition
correctly detected
falsely detected as transition,
labeled as meme
correctly detected
only one transition detected,
but labeled as two

(a) A true positive gradual transition

(b) One false negative transition in an animated slide

Source of meme video: Autodesk Research, "Exploring Generative 3D Shapes Using Autoencoder Networks", 2017, https://youtu.be/25xQs0Hs1z0

(c) A false positive transition in case of memes

Source of meme video: ML6, "Age interpolation in StyleGAN's latent space", 2019, https://youtu.be/x3pdf9S60zo

(d) One false negative transition due to fast slide change after meme insertion

Fig. 5: Qualitative results for slide transition detection using SliTraNet: Correct and failure cases.

approach with an F_1 score of around 81 % misses more slide transitions than the competing methods and due to the high complexity in the first two steps consumes a long execution time. In contrast, SliTraNet takes less than 90 min to process the 190 min test data.

Overall our SliTraNet demonstrates high effectiveness in the task of slide transition detection in lecture videos, which is also confirmed by the qualitative evaluation. In Fig. 5 some difficult cases are depicted to highlight the advantage of our method and also define some limitations. One difficulty is represented by animated slides, where little content changes in a short time. Fig. 5a shows an example of a correctly detected gradual transition, where the start and end point are marked by the blue arrow. From the hard transitions in Fig. 5b only the right one is detected by SliTraNet. A plausible reason for the failure of the network for the first transition is the small difference of the two frames as only thin lines appear that connect the nodes, while for the detected transition the slide change is larger due to the added node. Another difficulty that arises are the memes that are inserted into the lecture videos. The meme in Fig. 5c has a similar color distribution as the lecture slides and thus the transition within the meme is falsely detected as a slide transition. In Fig. 5d an example is shown, where the meme was inserted to the end of a static slide. The slide-video transition is correctly detected, but from the two fast slide changes, only one is detected. In the first step of the approach, we defined that a static slide has to be at least eight frames long, hence slides of one frame length cannot be detected by our method, but for the most applications these limitations are acceptable.

V. CONCLUSIONS

We presented a deep learning method to detect slide changes in lecture videos such as hard and gradual transitions. The quantitative evaluation showed a high performance of our method for this task. Future work could comprise extending the approach for a larger dataset and integrating it for online teaching, for instance to automatically insert slides for creating lecture notes in the AutoBlog framework.

REFERENCES

[1] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, "Dual Path Networks," *Advances in Neural Information Processing Systems (NIPS)*, vol. 30, pp. 4467–4475, 2017.

[2] P. Eruvaram, K. Ramani, and C. S. Bindu, "An Experimental Comparative Study on Slide Change Detection in Lecture Videos," *International Journal of Information Technology (IJIT)*, vol. 12, no. 2, pp. 429–436, 2020.

[3] Q. Fan, A. Amir, K. Barnard, R. Swaminathan, and A. Efrat, "Temporal Modeling of Slide Change in Presentation Videos," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, pp. 989–992, 2007.

[4] Q. Fan, K. Barnard, A. Amir, and A. Efrat, "Robust Spatiotemporal Matching of Electronic Slides to Presentation Videos," *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2315–2328, 2011.

[5] G. Gigonzac, F. Pitie, and A. Kokaram, "Electronic Slide Matching and Enhancement of a Lecture Video," *4th European Conference on Visual Media Production (CVMP)*, pp. 1–7, 2007.

[6] M. Guan, K. Li, R. Ma, and P. An, "Convolutional-Block-Attention Dual Path Networks for Slide Transition Detection in Lecture Videos," *International Forum on Digital TV and Wireless Multimedia Communications (IFTC)*, pp. 103–114, 2019.

[7] A. Hassanien, M. Elgharib, A. Selim, S.-H. Bae, M. Hefeeda, and W. Matusik, "Large-Scale, Fast and Accurate Shot Boundary Detection Through Spatio-Temporal Convolutional Neural Networks," *arXiv preprint arXiv:1705.03281*, 2017.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[9] A. Hernandez and S. H. Yang, "Multimodal Corpus Analysis of Autoblog 2020: Lecture Videos in Machine Learning," *Proceedings of the International Conference on Speech and Computer (SPECOM)*, pp. 262–270, 2021.

[10] H. J. Jeong, T.-E. Kim, H. G. Kim, and M. H. Kim, "Automatic Detection of Slide Transitions in Lecture Videos," *Multimedia Tools and Applications*, vol. 74, no. 18, pp. 7537–7554, 2015.

[11] H. Jiang, A. S. Helal, A. K. Elmagarmid, and A. Joshi, "Scene Change Detection Techniques for Video Database Systems," *Multimedia systems*, vol. 6, no. 3, pp. 186–195, 1998.

[12] H.-W. Kang and X.-S. Hua, "To Learn Representativeness of Video Frames," *Proceedings of the Annual ACM International Conference on Multimedia*, p. 423–426, 2005.

[13] O. Köpüklü, X. Wei, and G. Rigoll, "You Only Watch Once: A Unified CNN Architecture for Real-Time Spatiotemporal Action Localization," *arXiv preprint arXiv:1911.06644*, 2019.

[14] W. H. Leung, T. Chen, F. Hendriks, X. Wang, and Z.-Y. Shae, "eMeeting: A Multimedia Application for Interactive Meeting and Seminar," *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 2994–2998, 2002.

[15] R. Liang, Q. Zhu, H. Wei, and S. Liao, "A Video Shot Boundary Detection Approach Based on CNN Feature," *IEEE International Symposium on Multimedia (ISM)*, pp. 489–494, 2017.

[16] W. Liu, T. Mei, Y. Zhang, C. Che, and J. Luo, "Multi-Task Deep Visual-Semantic Embedding for Video Thumbnail Selection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3707–3715, June 2015.

[17] Z. Liu, K. Li, L. Shen, and P. An, "Sparse Time-Varying Graphs for Slide Transition Detection in Lecture Videos," *International Conference on Image and Graphics (ICIG)*, pp. 567–576, 2017.

[18] Z. Liu, K. Li, L. Shen, R. Ma, and P. An, "Spatio-Temporal Residual Networks for Slide Transition Detection in Lecture Videos," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 13, no. 8, pp. 4026–4040, 2019.

[19] J. Luo, C. Papin, and K. Costello, "Towards Extracting Semantically Meaningful Key Frames From Personal Video Clips: From Humans to Computers," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 19, no. 2, pp. 289–301, 2009.

[20] D. Ma and G. Agam, "Lecture Video Segmentation and Indexing," *Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIRE), Document Recognition and Retrieval XIX*, pp. 238–245, 2012.

[21] A. Maier, "AutoBlog," https://autoblog.tf.fau.de/, 2020.

[22] A. Mavlankar, P. Agrawal, D. Pang, S. Halawa, N.-M. Cheung, and B. Girod, "An Interactive Region-Of-Interest Video Streaming System for Online Lecture Viewing," *18th International Packet Video Workshop (PV)*, pp. 64–71, 2010.

[23] D. Perelman, "Slide-detector," https://git.aweirdimagination.net/perelman/slide-detector, 2020.

[24] G. Schroth, N.-M. Cheung, E. Steinbach, and B. Girod, "Synchronization of Presentation Slides and Lecture Videos Using Bit Rate Sequences," *18th IEEE International Conference on Image Processing (ICIP)*, pp. 925–928, 2011.

[25] T. Souček and J. Lokoč, "TransNet V2: An Effective Deep Network Architecture for Fast Shot Transition Detection," *arXiv preprint arXiv:2008.04838*, 2020.

[26] T. Souček, J. Moravec, and J. Lokoč, "TransNet: A Deep Network for Fast Detection of Common Shot Transitions," *arXiv preprint arXiv:1906.03363*, 2019.

[27] T. Wang, N. Feng, J. Yu, Y. He, Y. Hu, and Y.-P. P. Chen, "Shot Boundary Detection Through Multi-stage Deep Convolution Neural Network," *International Conference on Multimedia Modeling (MMM)*, pp. 456–468, 2021.

[28] X. Wang and M. Kankanhalli, "Robust Alignment of Presentation Videos with Slides," *Pacific-Rim Conference on Multimedia (PCM)*, pp. 311–322, 2009.

[29] B. Zhao, S. Lin, X. Qi, R. Wang, and X. Luo, "A Novel Approach to Automatic Detection of Presentation Slides in Educational Videos," *Neural Computing and Applications*, vol. 29, no. 5, pp. 1369–1382, 2018.

[30] B. Zhao, S. Lin, X. Qi, Z. Zhang, X. Luo, and R. Wang, "Automatic Generation of Visual-Textual Web Video Thumbnail," *Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH) Asia Posters*, pp. 1–2, 2017.