Automated Video Edition for Synchronized Mobile Recordings of Concerts

Albert Jiménez[1][a], Lluís Gómez[2][b] and Joan Llobera[1][c]
[1]*IZI C/casp 40 Ppal 1, 08010 Barcelona, Spain*
[2]*Computer Vision Center, Campus UAB, Edifici O, 08193 Cerdanyola del Vallès, Spain*

Keywords: Automated Video Edition, Computer Vision, Synchronized Recordings, Multi-camera Recordings, Camera Selection, Attention Mechanism, Pointer Networks.

Abstract: We propose a computer vision model that paves the road towards a system that automatically creates a video of a live concert by combining multiple recordings of the audience. The automatic edition system divides the edition problem in three parts: synchronize recordings with media streaming technology, selection of the scene cut position, and the selection of the next shot among the different contributions using an attention-based shot prediction model. We train the shot prediction model using camera transitions in professionally-edited videos of concerts, and evaluate it with both an accuracy metric and a human judgement study. Results show that our system selects the same video source as the ground truth in 38.8% of the cases when challenged with a random number of possible sources ranging between 5 and 10. For the rest of the samples, subjective preference among the selected image and the ground truth is at chance level for non-experts. Image editing experts do reflect better-than-chance performance, when asked to predict the following shot selected.

1 INTRODUCTION

The abundance of mobile phone recordings in live events creates an opportunity for a collaborative approach to video creation, where each member of the audience records whatever fragments she wants to, and an automated solution combines all the recordings to create a video showing the entire event from multiple angles and perspectives. Here we present a computer vision model that paves the road towards an automated system that combines the different video recordings of an audience and the audio recorded by the event organizer to create a common video edit. More specifically, given a set of videos recorded by different people and streamed synchronously to a server, the computer vision model cuts among different shots and selects the best cut transitions among the options available (see Figure 1). The result is a video that combines different viewpoints, but whose recordings are still synchronized with the audio recorded in the event. Our solution separates the automatic edition problem in three parts: synchronize recordings with media streaming technology, select the moment where there is a scene cut, and select the next shot among the available contributions. In this paper we focus on the later module: next shot prediction. We present a model that learns to select the most suitable camera from all the cameras available at any given

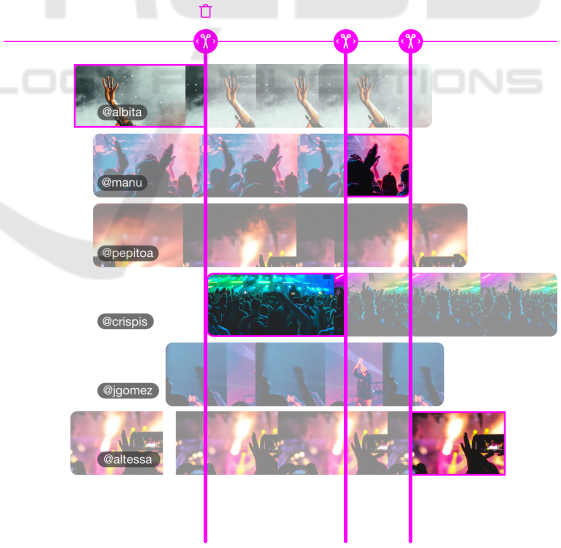


Figure 1: Given a set of synchronised videos and cut positions (vertical lines), the proposed model chooses automatically the next shot.

[a] https://orcid.org/0000-0002-3482-7589
[b] https://orcid.org/0000-0003-1408-9803
[c] https://orcid.org/0000-0002-9471-1334

Jiménez, A., Gómez, L. and Llobera, J.
Automated Video Edition for Synchronized Mobile Recordings of Concerts.
DOI: 10.5220/0010847600003124
In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 4: VISAPP*, pages 941-948
ISBN: 978-989-758-555-5; ISSN: 2184-4321
Copyright © 2022 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

time, inspired by Pointer Networks (Vinyals et al., 2015). Contrary to previously published works on automated video edition of multiple-camera concert recordings (Laiola Guimaraes et al., 2011; Shrestha et al., 2010), our model learns only from visual data and does not rely on heuristic rules or manually annotated meta-data.

The main contributions of this paper are: (1) a novel computer vision model that, when given multi-camera synchronized video streams, predicts the camera selected in the next cut; (2) a framework to obtain potentially unlimited training data for our model from existing edited videos; (3) extensive experiments conducted to demonstrate the validity of the proposed model. [1]

2 RELATED WORK

2.1 Automated Video Edition

Automated video edition in the context of multiple-camera recordings has been studied previously. (Laiola Guimaraes et al., 2011) proposed a semi-automatic method based on video annotations and a video selection algorithm aware of user preferences and video authors. In (Shrestha et al., 2010), concert mashup videos were generated by solving an optimization problem, maximizing the degree of fulfillment of several cinematographic requirements such as diversity or suitability of cut points from low-level features. A similar social multi-camera setup was addressed by (Arev et al., 2014) in the sports domain. Leveraging the insight that cameras focus their attention towards important content, they maximized the coverage of important content while adopting cinematographic guidelines such as the avoidance of jump cuts or the compliance of the 180-degree rule. These approaches depend on priors, annotations or low-level features and some require high computation capabilities for longer videos.

Other automated editing methods are based on complementary information that is provided along with the videos, such as a transcript in dialogue-driven or talking-head videos (Berthouzoz et al., 2012; Fried et al., 2019; Leake et al., 2017), an oral or written narration of the events (Truong et al., 2016; Wang et al., 2019) or the music clip in a music-driven video montage (Liao et al., 2015). It is not the case of our proposed solution, in which the edition is solely based on the video content. Also, some solutions base the editions on pre-defined editing idioms that the user can select (Leake et al., 2017; Liao et al., 2015; Wu and Christie, 2015), as opposed to our data-driven approach where stylistic choices are learned. A few data-driven solutions have also been reported. (Chen et al., 2018) propose a method for camera selection in soccer broadcasting in which the importance of video sequences is predicted with a random forest and C3D (convolutional 3-dimensional) features. (Wu and Jhala, 2018) extract audio and human pose features to automatically edit videos of corporate meetings. A Long-Short Term Memory neural network trained on features from professionally-edited videos is then used to predict joint attention and make edition decisions.

Opposed to the methods above, our solution detaches the shot duration and camera selection decisions.Once the transition time is determined, the most suitable camera is selected based on an attention mechanism trained on professionally-edited videos.Hence, in our novel approach each editing decision is independent from previous footage.

2.2 Input Selection based on Pointer Networks

A key element of our automated video editing system is the ability to select the most suitable camera from all the cameras available at any given time. For this we use a pointer mechanism inspired by pointer networks (Vinyals et al., 2015). Pointer networks are sequence-to-sequence models, where each token in the output sequence corresponds to a token at a certain position in the input sequence. The model selects (points to) an input token through an attention mechanism that models the probability distribution over the input tokens.

Pointer networks have been previously used in a variety of natural language processing and computer vision tasks – such as document summarization (See et al., 2017), neural machine translation (Gulcehre et al., 2016), or scene text visual question answering (Singh et al., 2019; Gómez et al., 2020) In this work we use a many-to-one architecture with a conditional attention mechanism. The model selects one of the many available inputs (cameras) conditioned on the previous camera frame. Furthermore, we also evaluate different types of attention (additive, multiplicative or scaled dot-product attention).

[1] An example of video generated with this method can be found at https://www.youtube.com/watch?v=GaO3lzVZbF0&ab_channel=IZIRecordingTogether

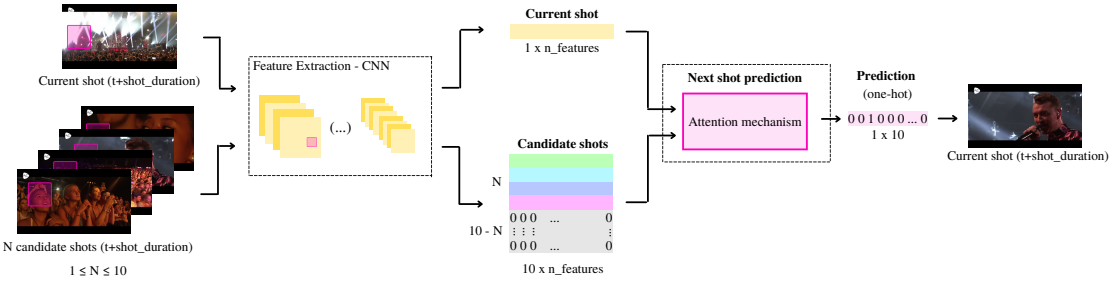


Figure 2: The next shot is predicted from the set of candidates at the transition time conditioned to the current shot.

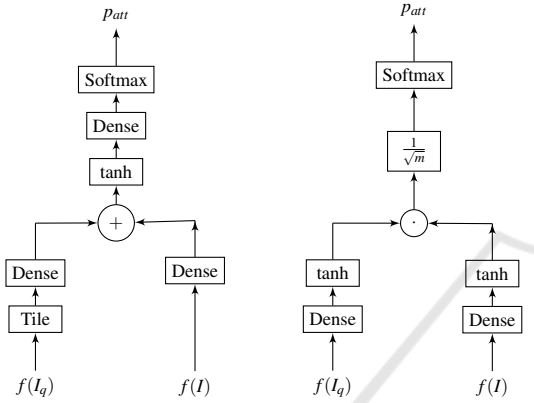


Figure 3: Computation graph of the attention mechanism f_{Att} for additive (a) and scaled dot-product (b) attention.

3 METHOD

Our solution is based on, first, using streaming technologies to capture the different camera recordings with a common timeline. Then, we determine *when* to cut and, finally *what* the next shot should be (see Figure 2). In this article we focus exclusively on the next-shot prediction module. The following-shot prediction module selects the best candidate among all available footage at that particular time instant. If no other camera is available, the camera selected remains the same. This strategy is similar to how a producer would switch cameras in a live broadcast, and allows for quick decisions based on a very limited amount of images which can be done on the fly, with no need to wait until the whole footage is available. Also, the computation time grows linearly with the duration of the final video, allowing to provide quick editions for both short and long events.

3.1 Next Shot Prediction

To select *what* shot better follows the one that is currently shown we use an attention mechanism fed with deep convolutional features obtained by processing the current frame with a pretrained CNN, and keeping the output of the last convolutional layer after a Global Average Pooling. We used both an Xception (Chollet, 2017) network pretrained on Imagenet[2] and a ResNet50 (He et al., 2016) pretrained on the Places365 (Zhou et al., 2017) data set to produce feature vectors of dimensionality 2048. A comparison of their performance when used alone and combined can be found in the experiments section. Figure 3 illustrates the computation graph of the attention mechanism that performs this task for two of the three attention types we compared: additive, dot-product and scaled dot-product. When we use additive attention, we first tile the features of the current frame $f(I_q)$ to match the dimensions of the candidate frame features $f(I)$ of shape $10 \times n_{features}$, where 10 is the maximum number of possible frame inputs during training and $n_{features}$ is the length of the feature vectors provided by the pretrained CNN (2048 in our case). Second, the tensors go through two different fully-connected layers with m output units before being added and activated with an hyperbolic tangent activation. The resulting tensor goes through a final dense layer of 10 units and a softmax activation function to generate the output attention vector p_{att}. This vector of size 1×10 represents the probability of each candidate frame to succeed the reference one, and the camera change is decided by sampling from this output distribution, instead of taking *argmax*, in order to get a more diverse and less "loopy" behaviour from the system. When we use either dot-product attention or scaled dot-product attention, both the features from the current frame $f(I_q)$ and candidate frames $f(I)$ go through a dense layer with m output neurons and an hyperbolic tangent activation function. Then, the dot product between the resulting tensors of size $10 \times m$ and $1 \times m$ respectively is calculated by stacking the dot-product between each row of the first tensor and the second tensor. For the case of scaled-dot product, the resulting tensor of size 1×10 is scaled by

[2]https://keras.io/api/applications/xception/

a scale factor $\frac{1}{\sqrt{m}}$ before applying a softmax activation to produce the output attention vector p_{att}. This is skipped in the case of dot-product, where no scale factor is applied. We treat the number of neurons on the hidden layers m of as a hyperparameter and tune it with a grid-search, as reported in section 4.3. We also observe that, when using Imagenet and Places365 features in combination, we get better results by fusing the features inside the model rather than just concatenating the feature vectors at the beginning. The model takes the query features from both data sets separately, learning different weights in the first dense layer until both tensors are concatenated right before the sum or dot-product step, depending on the attention type. The same applies to the features from the candidate frames.

4 DATASET AND TRAINING

The ideal dataset to train our model would be a set of images taken from different synchronized cameras. Since we could not find such a data set for live concerts, we generated a data set that approximates this ideal data set from edited live concert videos available on YouTube.

4.1 Dataset Generation

First, we manually selected and downloaded 100 professionally edited videos of live music performances involving several cameras. The videos range from 2 to 102 minutes in length. The videos depict different indoor and outdoor locations, different times of the day, and different musical styles so that our attention mechanism can learn to operate in different scenarios. Second, we process the videos in two sub-steps:

1. We detect scene cuts in the videos and extract a frame before and after each cut. The frame before the transition is going to be the query in our attention mechanism, and we will call it *current shot*. We consider the frame after the cut as the correct prediction or ground truth annotation.
2. We randomly sample frames from the same video, as well as from other videos to simulate the content of the other synchronized cameras that are recording the concert. As they must not be selected by our attention mechanism, we will name them *distractors*.

At training time the ground truth was shuffled with the distractors to form a set of images from which the attention mechanism will have to choose one given the current shot query. To extract the current shot and the ground truth we used a simple threshold-based scene change detector, PySceneDetect[3] (see Figure 4)The content-aware scene detector in PySceneDetect finds areas where the difference between two subsequent frames exceeds the threshold value that is manually set. Since using the default sensitivity threshold of 30.0 did not provide good results for all videos, when a significantly low number of transitions were found, the threshold value was lowered; when a significantly high number of transitions was found, we manually checked for false detections and increased the threshold until the number of false detections was minimal, even if some real transitions were lost in the process. Once a reliable list of transitions was generated for each video, we extracted the current shot and ground truth with a simple rule: the current shot is the image 5 frames before the transition, and the ground truth is the image 5 frames after. This 5 frames margin allowed us to avoid smoother transitions, where a shot faded into the next one as opposed to an immediate cut.

Distractors simulate synchronized shots at the time of the transition. They play an important role in the training and evaluation of the model, since they are the shots that the attention mechanism must learn to not select. As our model is intended to work with inputs of variable length, each pair of current shot and ground truth frames were related to a variable number K of distractors, ranging from a minimum of 4 to a maximum of 9. For each shot, half of these distractors were random frames from the same video where the current frame and the ground truth were extracted. The remaining distractors come from random frames of random videos other than the one of the current frame and ground truth. We expected the distractors extracted from random videos to be easier to discriminate but also important, as we wanted the attention mechanism to never choose videos with content inconsistent with the event being recorded. In addition, since all distractors were extracted from professionally edited videos, the large majority correspond to good quality images, both in terms of camera position and of stability. To reflect the fact that in our application scenario we expect to have low-quality images, we applied a random combination of vertical flip, image rotation and motion blur to a 10% of the distractors. Rotations were bounded between -20 and 20 degrees. Motion blur was applied as a filter either along the vertical, horizontal or diagonal directions, with a size of randomly set between 5 and 30. To generate the distractors for a pair of current frame and ground truth a feature vector was extracted for each distractor and shuffled with the feature vector of

[3]http://pyscenedetect-manual.readthedocs.io/

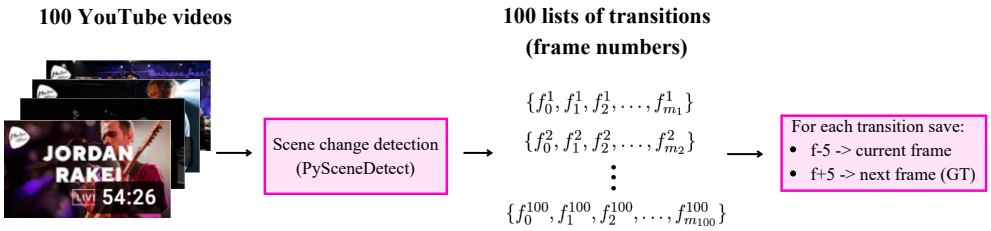


Figure 4: Extraction of the current frame and ground truth in the process of generating our data set.

the ground truth If the number of distractors K was lower than 9, we added $9-K$ zero vectors as padding to provide the attention mechanism with an input of constant size $10\times$ n_features.

4.2 Dataset Split

We split the 100 collected videos into three sets of 70, 10, and 20 for training, validation, and test respectively. To keep the number of transitions balanced, we ranked the videos by number of transitions, and sampled uniformly the ordered video list to obtain the validation and test sets. The remaining videos were considered the training set.

Adding distractors to generate the input vectors is handled differently in the training set, as opposed to the validation and test sets. At training time, for a given training sample, we pick a random number of new distractors for each epoch, pooling them from the set of pre-computed distractors. By doing so, the model learns to choose the best shot among many different options. As such, we increase the generalization power and reduce the chances of early overfitting. Opposite to this strategy, in the validation and test sets we added a fixed random number of distractors to each sample. We also use the same combination of sample and distractors for all our experiments. This is important to obtain evaluation metrics that do not depend on random factors that may vary at each iteration.

4.3 Training

We implemented our models using the Tensorflow[2.2.0] deep learning framework, and trained them with a NVIDIA GeForce RTX 2070 SUPER. For each combination of features (Imagenet, Places365, Imagenet+Places365) and attention type (additive, multiplicative, scaled dot-product) we conducted a grid search to optimize the validation accuracy by tuning the number of outputs of hidden layers (256, 512, 1024, 2048, 4096), the epochs (from 1 to 30), the Batch size (16, 32, 64, 128, 256), the Optimizer (Adam, Nadam) and the Learning Rate (0.0005, 0.001, 0.005).

5 EXPERIMENTS

To evaluate the performance of our next-frame prediction model we perform two groups of experiments. First, we evaluate the model using a standard accuracy metric and compare it with several baselines. Second, we perform a human judgement study to validate this solution with subjective metrics.

Table 1: Accuracy comparison of different models using random selection baselines, features pretrained on ImageNet (IN), features pretrained on Places365 (PL) and features pretrained on both IN and PL.

Method	IN	PL	Acc.
Random-10			0.100
Random-7.5			0.133
Random-SameVideo			0.250
Additive attention	✔	✗	0.293
Multiplicative attention	✔	✗	0.283
Scaled dot-product attention	✔	✗	0.312
Additive attention	✗	✔	0.358
Multiplicative attention	✗	✔	0.366
Scaled dot-product attention	✗	✔	0.388
Additive attention	✔	✔	0.361
Multiplicative attention	✔	✔	0.346
Scaled dot-product attention	✔	✔	0.380

5.1 Accuracy Metrics

We evaluated our model with an accuracy metric as in a standard classification task. We considered a prediction correct when it matched the ground-truth, and incorrect in any other case. We also included the following random and heuristic baselines:

- **Random-10** the expected accuracy of a model that selects one of the 10 inputs frames at random.

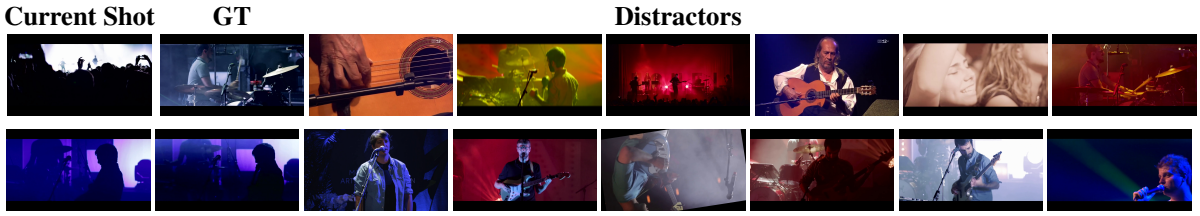


Figure 5: Test samples + distractors. On top the Current Shot and GT represent the images before and after a transition. Bottom, the Current Shot and GT are not images before and after a transition.

- **Random-7.5** the accuracy of randomly selecting one of the input frames that are not part of zero-padding. In average there are 7.5 non-zero inputs in our test set (see section 4.1).
- **Random-SameVideo** same as above but in this case randomly selecting a frame from the same video as the reference frame (query).

Table 1 shows a comparison of the obtained accuracies with these baselines and different variations of our model. In particular, we compare three different scoring functions for the attention mechanism (additive, multiplicative, and scaled dot product attention) using pretrained visual features on two different data sets (ImageNet and Places365). Our model's performance is above the defined baselines for any combination of features and attention type. It can be seen, however, that using pretrained features on Places365 represents a substantial improvement of at least 0.06 in accuracy with respect to using ImageNet features, while the combination of both does not lead to an improvement of the model's performance. Scaled dot-product attention, on the other hand, leads to the highest accuracy for any of the given features with a margin of at least 0.019. Hence, the attention mechanism that provides the best results is the one that uses scaled dot-product attention and pretrained features on Places365; with an accuracy of 0.388, the model achieved this performance with a Batch Size of 128, Nadam optimizer, a Learning Rate of 0.0005, hidden layers of size $m = 2048$ and 15 epochs. It is also well above any of the three baseline values.

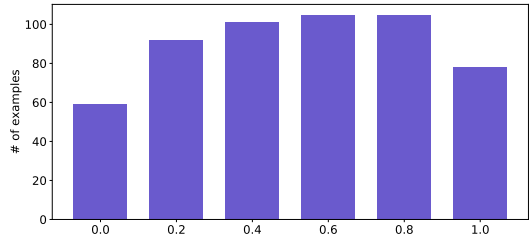
5.2 Human Judgement Study

Accuracy metrics have a clear drawback for our objective: they do not take into account the subjective nature of the task. Given the same query and set of inputs, different editors may not agree in which is the best next frame choice. To evaluate the quality of our best model (scaled dot-product attention using pretrained features on Places365) in a more realistic way we complemented the results with a human judgement study. Subjects were presented with a query image (current frame) from the test set and two options for the next shot: one option was the ground truth frame and the other option was the prediction of our model. They were asked to select the best option following these exact instructions: *In this task you will see a reference image from a live music video, your job is to select which of the two images below (A or B) you think is better as a transition (after the reference image) for a good scene cut*. The two options (ground-truth and prediction) were randomly assigned to option A or B.

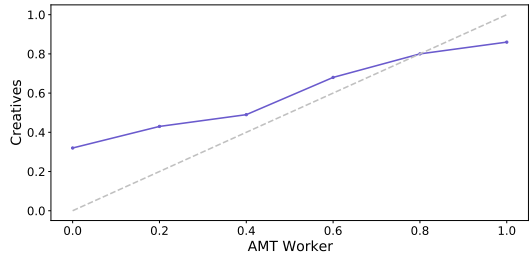
The study was conducted with a random subset of 540 examples from the test set, excluding the examples in which our model prediction matched the ground-truth. The rationale for this design was to assess if the distractors selected by our system were also perceived as ambiguous by humans, or rather the criteria used by the trained system was very different from subjective judgements. To measure the reliability of the responses from human subjects, we also added 60 control examples in which one of the options was a random image from the Places365 data set. We used two groups for this subjective study:

1. Professional Workers with a level of Master competence in Amazon Mechanical Turk(AMT), paid 0.05$ per response. In this group we collected 5 answers for each example.
2. A group of Creative Professionals in the fields of video edition and graphical design, recruited as volunteers among our group of acquaintances. In this group we collected a single answer for each example.

Both groups did the task in the AMT framework. A total of 32 AMT Workers and 4 Creative Professionals participated. For the AMT group, the ratio of correct answers in the control set was 0.993, and the average time dedicated per assignment was 1 minute and 44 seconds. For the Creative group, the ratio of correct answers in the control set was 1, and the average time dedicated per assignment was 28 seconds. AMT workers preferred the predicted frame (opposed to the GT) in 47,4% of the cases. When analyzed with majority voting, the predicted frame preference



(a) Histogram of images as preferred by AMT Workers.



(b) Relation between responses of AMT Workers and Creatives. The dotted line of slope 1 is used as a reference.

Figure 6: Subjective preferences of AMT workers per response accuracy. Accuracy is considered per group: accuracy of 1 occurs when the 5 responses of AMT Workers selected the ground truth option. Accuracy of 0 occurs when all responses selected the predicted frame.

was 46.67%. Professionals selected our prediction in 38.89% of the examples. Figure 6a shows the distribution of the collected answers for the 540 examples. We appreciate that in 59 of the examples all 5 workers selected the frame predicted by our model instead of the ground-truth frame (5 pred vs. 0 GT). Figure 6b shows the relation between the accuracy of AMT Workers and Creatives at selecting the GT frame, reflecting a strong correlation between the criteria of AMT Workers and Creative Professionals.

To further understand why Subjects clearly diverge from ground truth, we look into the images that belong to each of the accuracy ratios. Figure 7 shows a example of the images presented to the subjects for each possible response that we obtained, hand-picked to try to be representative of the group.

6 DISCUSSION

Accuracy tests show that our prediction model performs significantly better than chance. There is room for improvement, though: when we look into pairs where Subjects always preferred the GT over the Predicted image (see Figure 7, top two rows) the reason for Subjects to prefer the GT clearly seems that the Predicted image is of poor quality or does not follow the reference image as well as the GT image does.

Figure 7: Results of the human-judgement study. One example shown for each possible outcome. Top is 0 predicted vs 5 ground truth, below is 1 vs 4, until at the bottom which shows an example of 5 predicted vs 0 ground truth

However, taken globally, subjective tests confirm that when the Ground Truth and the Predicted image do not match human subjects (or, at least, AMT Workers) have an overall confusion rate of $47,4\%$. The fact that the majority preference is very close to this value (46.67%) also suggests there is a quite wide consensus on this fact (i.e., this ratio is not biased by one rogue subject). It is also true that the Predicted images that AMT Workers choose instead of the Ground Truth are significantly correlated with the preferences of Creative Professionals (Figure 6b). From this perspective, it would seem that for human subjects our automated method cannot be distinguished from the ground truth.

Further analysis, though, nuances this response, and we believe the reason is because not all distractors are created equal. If Subjects were not able to differentiate between Ground Truth and Predicted images for any example, most images would be in the central bins in Figure 6a. In those cases the responses of AWT Workers and of Creative Professionals match the most. It also seems that in these cases the GT and the Predicted image indeed are very similar (See central rows in Figure 7). However, the distribution across bins in Figure 6a is rather uniform. The highest divergence in responses occurs in the leftmost bins,

where Creative Professionals select the GT more often than AMT Workers. The examples in the two bottom rows of Figure 7) suggest that AMT Workers diverge from GT based on consistency of color or image composition. And overall, the global responses of Creative Professionals are biased towards the Ground Truth (38.89% versus 47,4%). We believe these two discrepancies can be best explained due to a difference in criterion, for particular cases: in examples when one of the two options shows a similar camera angle and content as the reference image AMT Workers tend to select it, while Creative Professionals usually choose the one that provides more diversity of shots.

Further work exploring automatic viewpoint analysis should be done to clarify this possibility, and use it to improve the next-shot prediction module. Further work should also explore whether the combination of the shot-duration and the next-shot prediction produces results that are more or less consistent with subjective preferences. Further directions to explore are to take advantage of features from the audio for both modules, as well as to enrich the shot selection process.

In conclusion, subjective and objective metrics provide evidence that our next-shot prediction module performs reasonable predictions, quite consistent with the criteria of both AMT Workers and Creative Professionals. We also showed that the accuracy metric alone is not reliable, subjective metrics must also be considered.

REFERENCES

Arev, I., Park, H. S., Sheikh, Y., Hodgins, J., and Shamir, A. (2014). Automatic editing of footage from multiple social cameras. *ACM Transactions on Graphics (TOG)*, 33(4):1–11.

Berthouzoz, F., Li, W., and Agrawala, M. (2012). Tools for placing cuts and transitions in interview video. *ACM Transactions on Graphics (TOG)*, 31(4):1–8.

Chen, J., Meng, L., and Little, J. J. (2018). Camera selection for broadcasting soccer games. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 427–435. IEEE.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258.

Fried, O., Tewari, A., Zollhöfer, M., Finkelstein, A., Shechtman, E., Goldman, D. B., Genova, K., Jin, Z., Theobalt, C., and Agrawala, M. (2019). Text-based editing of talking-head video. *ACM Transactions on Graphics (TOG)*, 38(4):1–14.

Gómez, L., Biten, A. F., Tito, R., Mafla, A., Rusiñol, M., Valveny, E., and Karatzas, D. (2020). Multimodal grid features and cell pointers for scene text visual question answering. *arXiv preprint arXiv:2006.00923*.

Gulcehre, C., Ahn, S., Nallapati, R., Zhou, B., and Bengio, Y. (2016). Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Laiola Guimaraes, R., Cesar, P., Bulterman, D. C., Zsombori, V., and Kegel, I. (2011). Creating personalized memories from social events: community-based support for multi-camera recordings of school concerts. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 303–312.

Leake, M., Davis, A., Truong, A., and Agrawala, M. (2017). Computational video editing for dialogue-driven scenes. *ACM Trans. Graph.*, 36(4):130–1.

Liao, Z., Yu, Y., Gong, B., and Cheng, L. (2015). Audeosynth: music-driven video montage. *ACM Transactions on Graphics (TOG)*, 34(4):1–10.

See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

Shrestha, P., de With, P. H., Weda, H., Barbieri, M., and Aarts, E. H. (2010). Automatic mashup generation from multiple-camera concert recordings. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 541–550.

Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., and Rohrbach, M. (2019). Towards vqa models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8317–8326.

Truong, A., Berthouzoz, F., Li, W., and Agrawala, M. (2016). Quickcut: An interactive tool for editing narrated video. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 497–507.

Vinyals, O., Fortunato, M., and Jaitly, N. (2015). Pointer networks. *arXiv preprint arXiv:1506.03134*.

Wang, M., Yang, G.-W., Hu, S.-M., Yau, S.-T., and Shamir, A. (2019). Write-a-video: computational video montage from themed text. *ACM Trans. Graph.*, 38(6):177–1.

Wu, H.-Y. and Christie, M. (2015). Stylistic patterns for generating cinematographic sequences. In *4th Workshop on Intelligent Cinematography and Editing Co-Located w/Eurographics 2015*.

Wu, H.-Y. and Jhala, A. (2018). A joint attention model for automated editing. In *INT/WICED@ AIIDE*.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. (2017). Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.