Identifying Presentation Styles in Online Educational Videos

MSR-TR-2014-141

Anitha Kannan and Simon Baker

November 6, 2014

Abstract

The rapid growth of online educational videos has resulted in huge redundancy. The same underlying content is often available in multiple videos with varying quality, presenter, and presentation style (slide show, whiteboard presentation, demo, etc). The fact that there are so many videos on the same content makes it important to retrieve videos that are attuned to user preferences. While there are several aspects that drive user engagement, we focus on the presentation style of the video. Based on a large scale manual study, we identify the 11 dominant presentation styles that typically employed. We propose a reference algorithm combining a set of 3-Way Decision Forests with probabilistic fusion and using a large set of image, face and motion features. We analyze our empirical results to provide understanding of the difficulties of the problem and to highlight directions for future research on this new application. We also make the data available.

1 Introduction

The availability of educational videos on the web is growing rapidly. For instance, YouTube Education alone contains over 700,000 high quality educational videos from over 800 channels [13] such as the Khan Academy. Massive Open Online Courses (MOOCs) such as Coursera, EdX and Udacity are quickly gaining in popularity as radically new approaches to providing education. Research in the education literature has shown that the visual style of learning can effect content retention [19] and improve concept understanding [14], especially when paired with traditional course materials such as textbooks.

The proliferation of content has also resulted in massive redundancy. For instance, there are over 30 videos on YouTube on the topic of "the law of conservation of mass". For a user looking for a video on this topic, which one of these 30 videos would they prefer? This redundancy makes it important to account for user preferences during search while maintaining relevancy.

There are many facets to user preferences in the context of educational videos, including video quality, the nature of the presenter (e.g. is the speaker lively?) [12], and the presentation style (e.g. is the video a demonstration of the law, or is it an animation of it?). Understanding these facets allows algorithms to retrieve content that respect user preferences. Video search engines (e.g. Google or Bing Videos) might provide additional filters to re-rank results based on these preferences, analogous to faceted filters for image search [9]. We can also enrich the experience of a student learning from a digital form of a textbook in an electronic device with multimedia content augmentations that are attuned to their preferences [1, 12, 10].

The creators of educational videos use different styles in presenting the content. Examples include whiteboard lectures, video recordings of experiments, and slideshows. In this paper, we first identify all the major styles in a large scale study. We also propose a baseline algorithm and carefully analyze its perfor-

mance. The main contributions of our paper are:

- We introduce the application of inferring the presentation style of an educational video. We enumerate the 11 dominant presentation styles.
- We collect and label two datasets with different style distributions. We make these datasets available to the community to allow further progress.
- We propose a reference algorithm combining a set of 3-Way Decision Forest classifiers with probabilistic fusion and which use a diverse set of image, face, and motion features. Our results show the importance of independently learning maximally discriminative classifiers between pairs of classes (along with a background model), and fusing the results in a principled probabilistic manner.
- We present empirical results and ablation studies that highlight the practicality of our solution, the importance of various feature types, and suggest directions for future research.

1.1 Related Work: Web Video Categorization

The most closely related literature is the body of work on categorizing web videos [11, 15, 16, 18, 21, 23] into broad categories (e.g. humor, news, people and society, etc) based on the **content of the video.** The goal of this paper is to classify the **style of presentation** for a particular category of videos (educational videos). Our main contribution is introducing this new video classification problem, one with direct application to existing products such as video search engines, video portals, and online education providers.

Unlike the general video categorization problem, there is no pre-existing classification scheme or taxonomy of styles; identifying the set of valid styles, itself, becomes an important task. The styles of education videos are also less subjective than typical video categories such as humour or news. This leads to an easier labeling task and close to perfect inter-annotator agreement. The more focused scenario also makes it easier to develop more powerful features and better classifiers.

Figure 1: The 11 classes of education video.

2 Education Video Presentation Styles

By examining thousands of videos we identified 11 dominant presentation styles. These styles fall into two types, "rendered" videos (denoted "R") where the video is produced directly by a computer, and "videos" captured by a camera (denoted "V"). With this notation, the 11 presentation styles are (see Figure 1):

RS - Slideshow: The video is created with a slideshow tool and is a video of the slides.

RV - Slideshow with a Video of the Presenter: A video of the presenter added to the slides.

RA - Animation: These videos contain a computer generated animation, which can vary from quite simple to a "Hollywood" quality cartoon.

RP - Rendered Photographs: The video was created as a set of photographs, possibly with "Ken Burns" effect and overlaid text.

RH - Rendered Hand-drawn Slides: The slides are hand-drawn using a computer tool and stylus rather than using a real pen and paper.

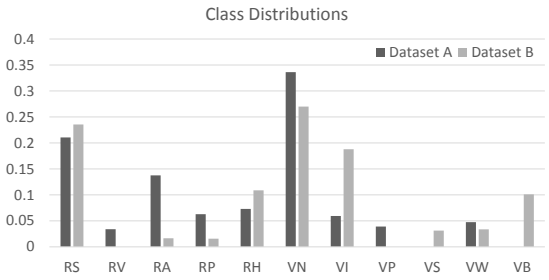

Figure 2: The class distributions of the two datasets.

VN - Natural Video: A "normal" video of some phenomenon, scene, or demonstration.

VI - Video of an Interview: A video of someone talking or explaining a concept.

VP - Video of Handwriting on Paper: A video of someone writing with a pen on paper.

VS - Video of Projected Slides: A video of a slideshow or slides projected onto a screen.

VW - Video of a Whiteboard: A video of someone in front of a whiteboard.

VB - Video of a Blackboard: A video of someone in front of a blackboard.

3 Curation of Labeled Datasets

We collected two data sets. Both consist of videos from YouTube that were specifically tagged as "education." They will be made available to other researchers along with their class labels. The two datasets were collected in slightly different ways leading to different distributions over the classes. See Figue 2.

Dataset A - Textbook Videos: This dataset of 589 videos was collected by considering a textbook and retrieving videos relevant to each section of the book using the COMITY algorithm [2]. This dataset captures variability in the presentation styles, when the content of videos correspond to a single theme. The ground-truth labels were generated by one of the authors.

Dataset B - Videos With Transcripts: We collected a second set of 1278 videos by considering all videos tagged 'education' that are available with a transcript. The presence of user-uploaded transcripts serves as a proxy to restrict videos to be truly educational content. This dataset captures the overall distribution of presentation styles in educational videos. The ground-truth labels were obtained using Amazon Mechanical Turk. As there are 11 classes to choose from, directly obtaining judgments using Mechanical Turk is challenging. Therefore, we obtained judgments in two phases: In the first phase, judges were asked to label if the video was predominantly a computer rendering (R) or a video recording (V). In the second phase, the videos were labeled for the (5 or 6) sub-classes within the main category.

4 Video Representation

We represent each video by three broad classes of features: "image" features i.e. features that can be computed for each frame independently,"face" features i.e. features that depend on detecting faces in the video, and "motion" features i.e. features that depend on how the video changes from frame to frame. We use 21 features in total, 6 image features, 6 face features, and 9 motion features.

4.1 Image Features

The presentation style is often very apparent from a single frame in the video. For example, the top row of Figure 3 includes one frame from a slideshow (RS) and one frame from a natural video (VN). Visually, the frames are very different. Distinguishing similar images (hand-drawn line images and natural images) is often performed by web image search engines, and exposed as faceted search filters [9].

Similar features can be used in videos. The features we use are based on the fact that regular photographs and rendered graphics typically have very different pixel and edge statistics [7, 17, 6]. For exmaple, in the second row of Figure 3 we include the intensity histograms computed by converting the images in the

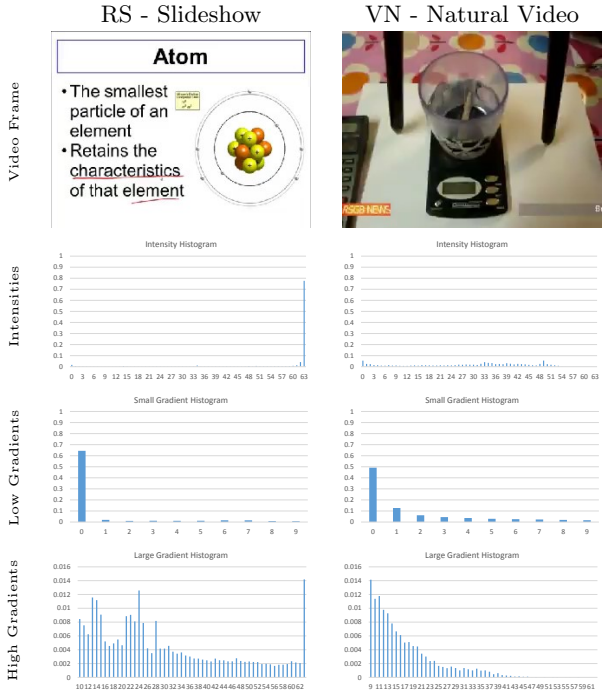

Figure 3: Our image features are based on the fact that the intensity and gradient histograms are often very different.

top row to greyscale and then binning the pixel intensities into 64 bins each consisting of 4 possible grey levels, $\text{bin}_0 0$=[0,3], bin_1=[4,7], ..., $\text{bin}_6 3$=[252,255]. The slideshow has a dominant bin which corresponds to the white background, whereas the natural video has a fairly uniform distribution across bins. To allow invariance to the grey level of the background, we first sort the bins by their values, from largest to smallest. Suppose that the i^{th} sorted intensity bin of frame f in the video has a weight $\text{IBinS}_i(f)$. We then compute the number of sorted bins required to fill a certain "contrast" threshold $\text{T}_{\text{contrast}}$ fraction of the pixels:

$$\text{Contrast}(f) = \min_l \left\{ l : \sum_{i=0}^{l} \text{IBinS}_i(f) \geq \text{T}_{\text{contrast}} \right\}. \quad (1)$$

We then compute a contrast feature by averaging this value across the video:

$$\text{feat}_{\text{contrast}} = \frac{1}{\#\text{frames}} \sum_{f=1}^{\#\text{frames}} \text{Contrast}(f). \quad (2)$$

There is also often a significant difference in the edge statistics across the classes [7, 17, 6]. In the 3rd a and 4th rows of Figure 3 we include histograms of the gradient magnitude for the images. We split the histogram so we can display the two parts at different scales. In the 3rd row we include the histogram for fairly weak edges. In the 4th row we include the part of the histogram for stronger edges. The slideshow image has relatively many zero gradients in the first bin due to the constant background, relatively few weak, but non-zero gradients, and relatively many very strong gradients due to the text and lines in the slide. Suppose $\text{GBin}_i(f)$ is the i^{th} gradient magnitude bin for frame f. We then define three more features: $\text{feat}_{\text{0-grad}}$ is the amount of weight on average across the frames in the zero gradient bin, $\text{feat}_{\text{low-grad}}$ is the amount of weight in the first few non-zero bins, and $\text{feat}_{\text{high-grad}}$ is the amount of weight in the strongest edge bins.

We also spatially estimate the amount of intensity noise in the video $\text{feat}_{\text{noise}}$. We fit a linear model to the pixel intensities in a 3×3 window and measure

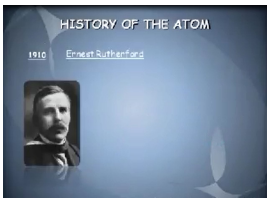

Figure 4: Sometimes images of faces appear in a slide show. To distinguish between faces in a slide and a video of a presenter, we check whether the face is moving.

the standard deviation of the error of the actual intensities from that model. Again, we average across the frames.

The 6 image features consist of 2 contrast features $\text{feat}_{\text{contrast}}$ for two different thresholds, $\text{feat}_{\text{0-grad}}$, $\text{feat}_{\text{low-grad}}$, $\text{feat}_{\text{high-grad}}$, and $\text{feat}_{\text{noise}}$.

4.2 Face Features

Some presentation styles prominently feature the face of the presenter, whereas others do not. We therefore introduce a face detection feature:

$$\text{feat}_{\text{face}} = \frac{1}{\#\text{frames}} \sum_{f=1}^{\#\text{frames}} \text{Face}(f) \quad (3)$$

$$\text{Face}(= \begin{cases} 1 & f \text{ has 1 face} \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

We use a multiple instance generalization [24] of the Viola-Jones algorithm [20].

This simple feature ($\text{feat}_{\text{face}}$) has a number of limitations. As illustrated in Figure 4, sometimes a slideshow (RS) might contain a face that is not that of the presenter. One solution to this problem is to check that the face is moving by computing the pixelwise difference between the current frame and the previous frame. Specifically, we check that the average difference across pixels is above a threshold. We denote this feature $\text{feat}_{\text{moving-face}}$. We also compute a second variant $\text{feat}_{\text{moving-face2}}$ by checking to see if the face box is moving (rather than the pixels in it).

A second problem is that face detection can be intermittent, particularly for slideshows containg a video of the presenter because the side of the face tends to be quite small. The combination of the small face, poor video quality, and pose and illumination changes can cause the face detector to fail on some frames. One solution to this is to compute the length of the longest segment of frames where no face is detected: $\text{feat}_{\text{face}*} =$

$$1.0 - \frac{1}{\#\text{frames-1}} \max_{l \le k} \{k - l : \text{Face}(f) = 0 \; \forall \; f \in [l, k]\} \quad (5)$$

So long as a face is detected every few frames the modified feature will be close to 1.0 and an intermittently failing detector will not be penalized much. Similarly, we compute the longest segment of frames where the face is always detected $\text{feat}_{\text{face}\dagger}$ which gives a sense of how stable the detection is. Finally, we compute the face size $\text{feat}_{\text{face-size}}$ as the square root of the average fraction of the area that is occupied by the face.

In summary, the 6 face features are $\text{feat}_{\text{face}}$, $\text{feat}_{\text{moving-face}}$, $\text{feat}_{\text{moving-face2}}$, $\text{feat}_{\text{face}*}$, $\text{feat}_{\text{face}\dagger}$, and $\text{feat}_{\text{face-size}}$.

4.3 Motion Features

There are a wide variety of possible motion features. Here, we break the possibilities into 3 types: (1) features that measure how often motion occurs, the frequency of motion, (2) features than measure how much of the image moves, the amount of motion, and (3) features that depend on the type of the motion.

4.3.1 Frequency of Motion

Some videos are always moving, whereas other videos just move once in a while. For example, animations (RA) are typically moving most of the time, whereas slideshows (RS) only move when there is a slide transistion. In Figure 5 we plot the motion magnitude across the frames in the video for an animation (RA) and a slideshow (RS). We compute the magnitude of the motion by first converting the frames to greyscale

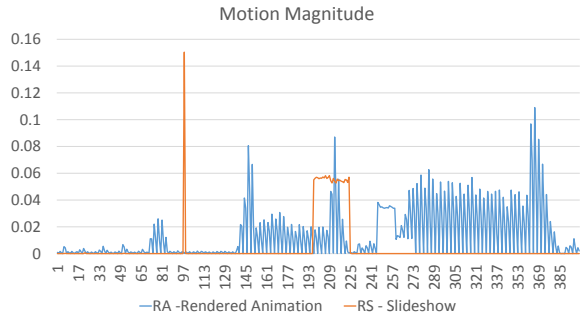

Figure 5: The frequency of motion varies considerably.

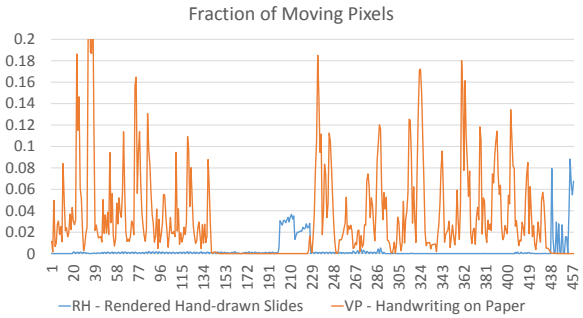

Figure 6: The amount of motion varies across classes.

and then computing:

$$\text{MMag}(f) = \frac{1}{256 \times \#\text{pixels}} \sum_{x,y} |I_{x,y}(f) - I_{x,y}(f-1)| \tag{6}$$

where $I_{x,y}(f)$ is the intensity of the grayscale pixel (x, y) of frame f. To distinguish these different types of videos, we introduce a motion frequency feature:

$$\text{feat}_{\text{motf}} = \frac{1}{\#\text{frames - 1}} \sum_{f=2}^{\#\text{frames}} \text{Mot}(f) \tag{7}$$

$$\text{Mot}(f) = \begin{cases} 1 & \text{MMag} \geq \text{T}_{\text{motf}} \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Like the face features, we add two variants computed by measuring the longest segment of frames where there is ($\text{feat}_{\text{motf}^*}$) or isn't ($\text{feat}_{\text{motf}^\dagger}$) motion.

4.3.2 Amount of Motion

Another thing that varies across videos is how much of the frame moves. For example, in a video of rendered hand-drawn slides (RH) a very small number of pixels will be changing in each frame, just the pixels being edited. In a video of a person writing on paper (VP), many more pixels will be changing because the moving hand is visible. See Figure 6. We compute whether each pixel is moving independently:

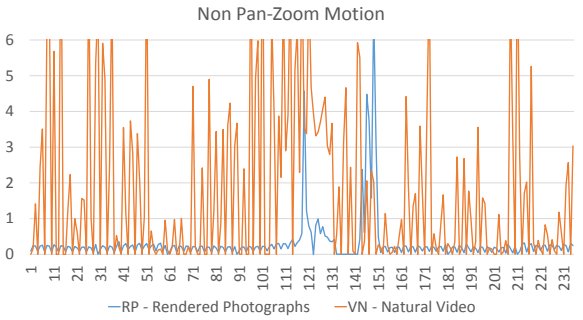

Figure 7: While some presentation styles (e.g. VN - Natural Videos) contain a lot of non-rigid motion, for others the motion is largely rigid (e.g. RP - Rendered Photographs).

$$\text{Mov}(f, x, y) =$$

$$\begin{cases} 1 & \text{if } |I_{x,y}(f) - I_{x,y}(f-1)| \geq \text{T}_{\text{motpix}} \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

and then compute the fraction of moving pixels:

$$\text{FracMov}(f) = \frac{1}{\#\text{pixels}} \sum_{x,y} \text{Mov}(f, x, y). \tag{10}$$

To make our features robust to extreme motions during transistions, etc, we compute: $\text{feat}_{\text{mota}} = \text{Percentile}_f(\text{FracMov}(f), \text{T}_{\text{mota}})$ where Percentile_f sorts the values $\text{FracMov}(f)$ and then chooses the value at the T_{mota} percentile.

4.3.3 Type of Motion

The type of motion varies across the presentation styles. For example, in a "Ken Burns" effect in a video of rendered photographs (RP), the motion might be a single "rigid" pan and zoom. On the other hand, in a natural video, there will likely be lots of different non-rigid components to the motion. We estimate the amount of non-rigid motion by first computing optical flow [8, 3], then estimating a rigid "pan and zoom" parametric motion [4] from the optical flow. Finally, we subtract the parametic motion from the optical flow and compute the magnitude across the frame. Denote the resulting non-rigid flow magnitude NRFlow(f). Figure 7 compares NRFlow(f) across frames for a video of rendered photographs (RP) and a natural video (VP). Like the amount of motion features, we use a percentile across frames to be robust to extreme motion during transitions. We compute $\text{feat}_{\text{mott}} = \text{Percentile}_f(\text{NRFlow}(f), \text{T}_{\text{mott}})$. We also compute a relative measure as the fraction of the optical flow magnitude (OFlow(f)) that is non-rigid: $\text{feat}_{\text{mott2}} = \text{Percentile}_f(\text{NRFlow}(f)/\text{OFlow}(f), \text{T}_{\text{mott2}})$. Finally, we also compute a feature $\text{feat}_{\text{mott3}} = \text{Percentile}_f(\text{OFRes}(f), \text{T}_{\text{mott3}})$ based on the Optical Flow residual (OFRes(f)). One thing that $\text{feat}_{\text{mott3}}$ captures is whether the changes in the video are due to motion (small optical flow residual) or due to the appearance and disappearance of scene elements (for example in a slideshow.) It also provides a second estimate of the noise in the video, like $\text{feat}_{\text{noise}}$.

In summary, the 9 motion features consist of 2 versions of $\text{feat}_{\text{motf}}$ for two different thresholds , two versions of $\text{feat}_{\text{mota}}$ for two different thresholds, and a single feature for each of $\text{feat}_{\text{motf}_*}$, $\text{feat}_{\text{motf}_\dagger}$, $\text{feat}_{\text{mott}}$, $\text{feat}_{\text{mott2}}$, and $\text{feat}_{\text{mott3}}$.

4.4 Processing Time

Feature extraction typically takes around 20 seconds for the 30 second video thumbnails in our datasets. To achieve this level of efficiency, some of the features are performed on a subsampling of the frames; for example, face detection is run on every 5th frame.

5 Presentation Style Identification

We pose the problem of presentation style identification as a classification task over the 11 dominant styles discussed in Section 2. We assume we are given a labeled data set such that each video is represented using the 21 features, $\mathbf{x}$ as described in Section 4, and its corresponding style, $y \in C$. Given such as labeled data set, we use an instantiation of 'stacked generalization' [22] that provides a rich framework for combining varied feature sets and classifiers for increased robustness and generalization. We first train multiple base classifiers. The outputs of these classifiers, which are now in the same space of prediction probabilities, are combined to learn the final classifier.

The choice for the base classifiers is based on the following observation: Multiple presentation styles have shared characteristics that overlap considerably; For example, both RV (slideshow with video of a presenter) and VI (video of an interview) videos have presenters in the video. Therefore, we would like base classifiers that systematically focus on regions of the discriminant surface between pairs of styles, while treating the remaining styles as noise.

Algorithm 1 shows the classification algorithm used in this paper. We first learn 3-way classifiers between pairs of styles and an additional background category ($\perp$) that consists of all styles other than the styles in the pair under consideration. $\perp$ captures the possibility that the true style can be different from the styles in the pair. For this, we divide the training data L into two non-overlapping subsets, B and S. We use B to train all the $\mathcal{K} = 11 \times 10/2$ 3-way classifiers. Once all are trained, each training sample $(\mathbf{x}, y) \in S$ is represented using $\mathbf{z}$ which consists of $3 \times \mathcal{K}$ features, the prediction probabilities from the $\mathcal{K}$ classifiers. These instances along with their labels are used to create a new training set L'_s which is used to train the final classifier H.

TrainClassifier: We trained the component classifiers, $H_{c_1 c_2}$ and H using Decision Forests [5]. A

Algorithm 1 Presentation Style Identification

Input: $L = \{(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^n, y^n)\}$ of n labeled instances, where $\mathbf{x}^j$ is the features of instance $\mathbf{x}^j$ and $y^n \in C$ is its style $C \in \{c_1 \cdots c_{11}\}$
Output: H, a classifier trained on L
Split L into B and S
/* Train a 3-way classifier for each pair of styles */
$H_{pool} = \emptyset$
for all $(c_1, c_2) \in$ ALLPAIRS(C) **do**
  $F = \{(\mathbf{x}, y) \in S \mid y = c_1 \text{ or } y = c_2\}$
  for all $(\mathbf{x}, y) \in S - F$ **do**
    $F = F \cup \{(\mathbf{x}, \perp)\}$
  end for
  $H_{c_1 c_2} =$ TRAINCLASSIFIER($\{(\mathbf{x}_I, y) \mid (\mathbf{x}, y) \in F\}$)
  $H_{pool} = H_{pool} \cup H_{c_1 c_2}$
end for
/* Use all 3-way classifiers to embed instances*/
/* in the space of style membership probabilities.*/
Define $L'_s = \emptyset$
for all $(\mathbf{x}, y) \in S$ **do**
  $\mathbf{z}' = \emptyset$
  for all $h \in H_{pool}$ **do**
    $\mathbf{z}^h =$ GETPREDICTIONPROBABILITIES($h, \mathbf{x}$)
    $\mathbf{z}' = \mathbf{z}' \cup \mathbf{z}^h$
  end for
  $L'_s = L'_s \cup \{(\mathbf{z}', y)\}$
end for
$H =$ TRAINCLASSIFIER(L'_s)

Decision Forest is an ensemble of D Decision Trees, $\{\mathcal{T}^i\}$, where each tree $\mathcal{T}^i$is independently trained using a random subset of feature-value combinations. During prediction, the output from each tree is combined to make the overall prediction for the forest. In particular, we used additive model for prediction (GETPREDICTIONPROBABILITIES) so that $p(c = c_j|\mathbf{x}, \{\mathcal{T}^i\}) = \frac{\sum_i p(c=c_j|\mathbf{x}, \mathcal{T}^i)}{D}$. Each tree is trained over a randomly chosen 25% of the features, with replacement and searched over all values of the features. Due to skewness in the data set, we balance it with repeated sampling with replacement. We use mutual information as the splitting criterion.

Parameters: Training is controlled by 3 parameters: the number of trees in the forest (200), the maximum depth of a tree (6), and the maximum imbalance when splitting a node (at least a 90–10 split.)

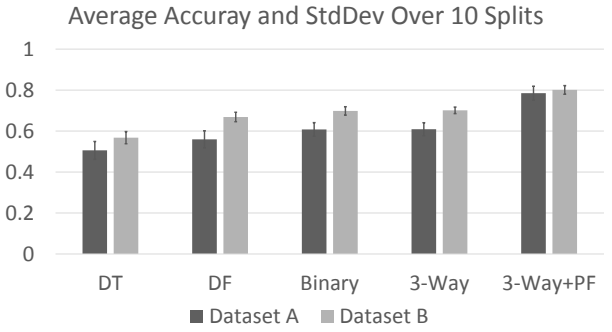

Figure 8: A comparison of our algorithm (3-Way+PF - a set of 3-way Decision Forests combined with probabilistic fusion) with several baselines. DT - a multi-class Decision Tree. DF - a multi-class Decision Forest. Binary - a set of binary Decision Forests combined with voting. 3-Way - a set of 3-way Decision Forests combined with voting.

6 Experimental Results

We begin by comparing our algorithm with several baselines: (1) a single multi-class Decision Tree, (2) a multi-class Decision Forest, (3) a set of 2-way (single style vs. all) binary Decision Forests combined using voting, (4) a set of 3-way (style A vs. style B vs. all) Decision Forests combined using voting, and (5) our algorithm consisting of a set of 3-way Decision Forests combined using probabilistic fusion. The Decision Trees and Forests are always trained in the same way (max depth 6, 200 trees per forest, at least a 90–10 split).

The results in Figure 8 show a consistent improvement as we move from a Decision Tree to a Decision Forest and on to a set of Binary or 3-Way Classifiers. The best results are obtained by combining a 3-Way Classifier with probabilistic fusion. We obtain around 80% accuracy on both datasets. The results show the importance of: (1) learning class-pair specific decision boundaries using Binary or 3-Way classifiers (rather than using a monolithic multi-way classifier such as a Decision Forest), and (2) the importance of combining the outputs of the base classifiers in a probabilistic manner (rather than by voting), something that can be formulated most naturally for a

	VN	RS	RA	RH	RP	VI	VW	VP	RV
VN	0.90	0.00	0.03	0.00	0.05	0.00	0.03	0.00	0.00
RS	0.00	0.80	0.04	0.00	0.12	0.00	0.00	0.00	0.04
RA	0.24	0.12	0.65	0.00	0.00	0.00	0.00	0.00	0.00
RH	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
RP	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
VI	0.14	0.00	0.00	0.00	0.00	0.86	0.00	0.00	0.00
VW	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
VP	0.40	0.00	0.00	0.00	0.00	0.00	0.00	0.60	0.00
RV	0.00	0.00	0.00	0.00	0.00	0.25	0.00	0.00	0.75

Dataset A - Videos corresponding to a textbook

	VN	RS	VI	RH	VB	VW	VS	RA	RP
VN	0.76	0.00	0.07	0.00	0.09	0.04	0.03	0.00	0.00
RS	0.02	0.85	0.03	0.05	0.00	0.00	0.00	0.03	0.02
VI	0.15	0.00	0.83	0.00	0.00	0.00	0.02	0.00	0.00
RH	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
VB	0.00	0.00	0.00	0.00	0.88	0.04	0.08	0.00	0.00
VW	0.00	0.00	0.11	0.00	0.11	0.67	0.11	0.00	0.00
VS	0.00	0.00	0.00	0.00	0.13	0.00	0.88	0.00	0.00
RA	0.20	0.40	0.00	0.00	0.00	0.00	0.00	0.40	0.00
RP	0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.75

Dataset B - Videos with Transcripts

Figure 9: Confusion Matrices for our 3-Way+PF algorithm.

3-way classifier.

We also studied the confusion matrices (see Figure 9) to understand the main sources of errors in discriminating the classes. We arrange the classes in the order of how frequently they appear in the two datasets. The last few classes in each case have relatively few examples and so their results should be taken as less significant.

We identified three main errors: First, RA is commonly confused with VN and RS (.65 vs. .24 and .12). This is not surprising as a slideshow (RS) with a lot of animations can be similar to an animation (RA), and a realistic animation can be very similar to a natural video (VN). Second, we find some confusion between whiteboard (VW), blackboard (VB), and videos of projected slides (VS). Again, this is not too surprising as visually they are quite similar, with a person presenting in front of a screen/board. It is possible we could do better at discriminating VW and VB if we added color features than distinguish light and dark regions better. Finally, VI is sometimes confused with VN, RS, and VW. Looking at the intermediate results, we found these errors to be caused largely by failures to detect low resolution faces in the video thumbnails.

6.1 Ablation Studies: Importance of Features

We also retained our 3-Way+PF algorithm on subsets of features to investigate how important the various feature are. In Figure 10 we include the results for Dataset B. Our findings carry forward to Dataset A and is omitted for space constraints. The classification results show that the motion features alone are the most powerful, but the removal of the face features handicaps the algorithm the most. These results indicate that the motion and image features are quite correlated and the inclusion of one can compensate for the omission of the other. The face features are more independent. Intuitively, this makes sense. There are certain style pairs that critically depend on face detection (VN vs. VI.) For other class pairs, a combination of features is needed to achieve optimal performance.

Studying the confusion matrices for the ablation studies (see the supplemental material) revealed the following: the image features are most important for VN (natural videos) vs. VB, VS, and VW (videos with a screen or board). The face features, as expected are most important for distinguishing VI from other classes that feature faces less prominently. Finally, the motion features are important for a large variety of class-pairs.

7 Conclusion

We introduced the problem of identifyin the presentation style of an educational video. We identified 11 major presentation styles and proposed an approach for inferring the particular style of an video based on image, motion and face features. We will be releasing the dataset to enable further research.

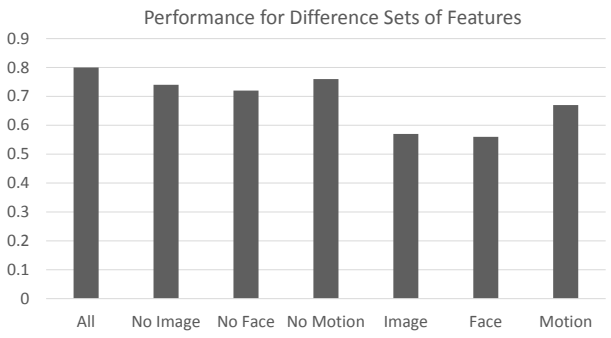

Figure 10: Top Left: Classification accuracy of our 3-Way+PF algorithm for different subsets of features.

We assumed that each video has is a single dominant presentation style (During ground-truthing we asked labelers to label what they felt was the dominate style.) This is not always the case. A video might start as an interview, and then shift to show an experiment. A natural next step, therefore, is to perform temporal segmentation and presentation style identification jointly. Another possible direction could be to use any additional signals such as the creator of the video, the audio track, or a transcript (when available.)

References

[1] R. Agrawal, M. Christoforaki, S. Gollapudi, A. Kannan, K. Kenthapadi, and A. Swaminathan. Mining videos from the web for electronic textbooks. Technical Report MSR-TR-2014-5, Microsoft Research, 2014. 1

[2] R. Agrawal, S. Gollapudi, A. Kannan, and K. Kenthapadi. Enriching textbooks with images. In *CIKM*, 2011. 3

[3] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 92(1):1–31, 2011. 7

[4] J. Bergen, P. Anandan, K.J.Hanna, and R. Hingorani. Hierachical model-based motion estimation. In *ECCV*, pages 237–252, 1992. 7

[5] A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7(2–3):81–227, 2012. 7

[6] R. Fergus, B. Singh, A. Hertzmann, S. Roweis, and W. Freeman. Removing camera shake from a single photograph. *ACM Transactions on Graphics*, 25(3):787–794, 2006. 3, 4

[7] D. Field. What is the goal of sensory coding? *Neural Computation*, 6:559–601, 1994. 3, 4

[8] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1–3):185–203, 1981. 7

[9] G. Hua and Q. Tian. What can visucal content analysis do for text based image search? In *ICME*, 2009. 1, 3

[10] M. Kokkodis, A. Kannan, and K. Kenthapadi. Assigning videos to textbooks at appropriate granularity. In *Educational Data Mining*, 2014. 1

[11] A. Kowdle, K.-W. Chang, and T. Chen. Video categorization using object of interest detection. In *ICIP*, 2010. 2

[12] S. Marioryad, A. Kannan, D. Hakkani-Tur, and E. Shriberg. Automatic characterization of speaking styles in educational videos. In *ICASSP*, 2014. 1

[13] M. Meeker and L. Wu. Internet trends. Technical report, KPCB, 2013. 1

[14] M. Miller. Integrating online multimedia into college course and classroom: With application to the social sciences. *MERLOT Journal of Online Learning and Teaching*, 5(2), 2009. 1

[15] C. Ramachandran, R. Malik, X. Jin, J. Gao, K. Nahrstedt, and J. Han. Videomule: a consensus learning approach to multi-label classification from noisy user-generated videos. In *Proc. of ACM Multimedia*, 2009. 2

[16] G. Schindler, L. Zitnick, and M. Brown. Internet video category recognition. In *Proc. of CVPR Workshop on Internet Vision*, 2008. 2

[17] E. Simoncelli. Statistical modeling of photographic images. In A. Bovik, editor, *Handbook of Image and Video Processing*, 2005. 3, 4

[18] Y. Song, M. Zhao, J. Yagnik, and X. Wu. Taxonomic classification for web-based videos. In *CVPR*, 2010. 2

[19] P. Tantrarungroj. *Effect of embedded streaming video strategy in an online learning environment on the learning of neuroscience*. PhD thesis, Indiana State U., 2008. 1

[20] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001. 5

[21] Z. Wang, M. Zhao, Y. Song, S. Kumar, and B. Li. Youtubecat: Learning to categorize wild web videos. In *CVPR*, pages 879–886, 2010. 2

[22] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992. 7

[23] S. Zanetti, L. Zelnik-Manor, and P. Perona. A walk through the webs video clips. In *Proc. of CVPR Workshop on Internet Vision*, 2008. 2

[24] C. Zhang and P. Viola. Multiple-instance pruning for learning efficient cascade detectors. In *NIPS*, 2007. 5