Image-Based Sentiment Analysis of Videos

Pankhuri Gupta, Balaji Soundararajan, Thomas Zachariah
Department of Computer Science and Engineering
University of Michigan
EECS 545 W14
{pankhuri, balijisb, tzachari}@umich.edu

Abstract

The work presented in this paper addresses the challenge of performing sentiment analysis on the visual features of video content. We use the method of Visual Sentiment Ontology (VSO) to extract Adjective Noun Pairs (ANP) and identify the sentiment score of each of the video frames. We then use HMM and SVM regression to identify the sentiment label of the entire video. We introduce a new method called local similarity-weighted scoring to improve upon the sentiment detection. Results for individual videos tested can be viewed interactively at http://umich.edu/~tzachari/545.

1 Introduction

There are a number of psychological studies that focus on testing in what ways videos evoke various emotions. Given as such, we believe sentiment analysis of videos is of great interest, and could provide further insight into what particular features in video elicit the corresponding emotional responses. This project addresses the task of detecting whether a video portrays a positive or negative sentiment. The model relies on detecting a set of visual concepts based on low level image features to infer the human-perceived sentiments portrayed by each frame of the video. Automatically assigning a sentiment score to a video clip poses significant challenges. The subjects, objects and background interact in complex ways to evoke an emotion. For instance, while a laughing man is a positive emotion, the emotion becomes negative when the same laughing man carries a weapon. We believe that well-trained models for detecting sentiments of images will capture such emotions. Additionally, the emotions within a clip vary with time and have a temporal sequence. We attempt to use HMMs and other methods in consideration of this. Additionally, we depict the test results of individual clips as a running plot to visually capture varying emotions throughout the video.

This work is divided into the following phases: Adjective Noun Pair (ANP) Detection, Sentiment Detection of Image and Video Processing. Our major contributions in this project are as follows:

1. We explore the use of a Naïve Bayes classification technique for **ANP Detection** Phase. Additionally, we experiment with multiple SVM regression settings to achieve the best possible results.
2. ***Application to Videos***: We extend the concepts in work by Borth et al. [1] to apply their image sentiment detection technique to videos on a frame-by-frame basis.
3. ***HMMs***: We form two alternate models of HMMs to calculate the sentiment score of frames of a video. This technique is applicable to our problem statement due to the presence of a temporal sequence of frames.
4. We propose a new method called ***Local Similarity-Weighted Score (LSWS)***, to

improve upon the sentiment scores of images. This method draws on the sequential nature of the frames in a video.

5. **Web Interface**: We present a web interface that gives the entire work that we have done as a part of this project. This interface can be released in the future.

2 Related Work

Sentiment analysis is a widely studied area, however, it has been limited to analysis of text data. Analyzing the sentiments of images is a relatively new field that is gaining more and more popularity with the social web [2] talks about using some very basic visual features and adjectives for finding sentiments portrayed by the images [1] introduces a concept of Adjective noun pairs that offer greater sentiments and uses a richer set of features. They train 1200 different binary classifiers (one for each ANP) and pass the test image through each of these classifiers. This gives a 1200 long vector, where each element gives the probability of corresponding ANP occurring in that image. They feed this vector as input to their Sentiment Detector binary classifier that labels the image as +1 or -1(negative).

We extend this work by applying the image sentiments to videos. Schaefer et al. [3] and Carvalho et al. [4], from whom we have obtained our testing data (see following section), refer to relatively recent psychophysiological studies on the direct human emotional response to video graphic imagery. Our intent with the application of image sentiment analysis to video, is to take first steps towards developing a model that can generate results comparable to those of such studies and to pinpoint the specific features responsible for various sentiments.

Hidden Markov Model assumes that the system is a Markov process with unobserved states. In Bilmes [5], the EM algorithm for HMM with Gaussian Models is described. Though HMMs are applied in temporal pattern recognition [6] such as speech [7], hand-writing etc., they have not been used to model the underlying sentiment of an image.

Deep Convolution Neural Networks have been recently shown to yield state-of-the art performance in challenging image classification benchmarks such as ImageNet [8]. While this classification deals with the problem of object recognition, it has not been applied for sentiment analysis. In this project, we have taken a step towards using CNNs for sentiment classification.

3 Dataset

Training ANP Detectors: The binary classifiers for detecting the presence of ANPs within an image are trained and tested using the Flickr Data [9] previously classified by the Visual Sentiment Ontology (VSO) [10]. The training data set comprises about 700 images per ANP. Libsvm's 5-fold cross validation is used for training purposes and an additional 20% of the data set is held out as validation set. The test data is divided into 5 parts and in total comprises about 300 images. The data sets are balanced and consists of equal number of positive and negative labelled examples.

Training Sentiment Binary Classifier: The data set that is used to train and test binary classifier for labelling images as positive and negative sentiment images is a set of 800 Twitter images provided by VSO [10]. This data set has an unequal number of images with negative sentiments. Hence, we have added 400 additional public domain images from Google.

Video Dataset: FilmStim database [3] and EMDB database [4] are used for running our models and testing our work. We have received permission for both the datasets to be used for research purposes.

Third-party Libraries Utilized:

1. The SVM trained binary classifiers for detecting the presence of ANPs in an image as provided by Visual Sentiment Ontology [10].
2. Scikit-learn: A python based library that provides the implementation for Gaussian HMM [11].
3. LibSVM: A Matlab library that implements various settings of a SVM [12].

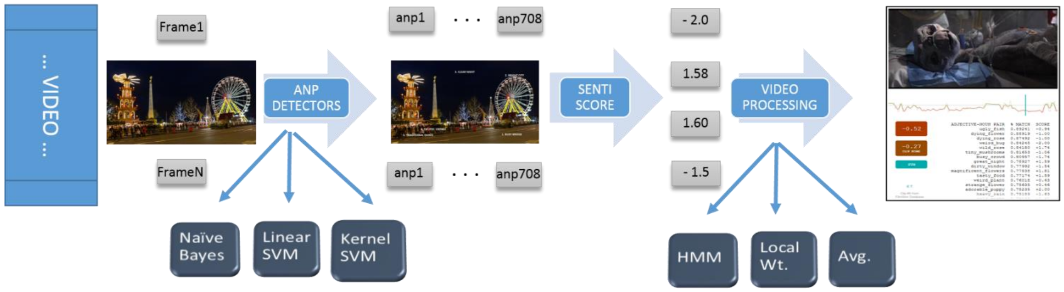

Figure 1: Overview of the proposed framework for constructing the visual sentiment ontology, SentiBank and Video Processing.

4 Methodology

The general process framework is depicted in the pipeline shown in Figure 1. The methods that we have utilized throughout the project are discussed in the following subsections.

4.1 ANP Detection Methods

4.1.1 Comparing Multiple SVMs *(Original Work)*

As mentioned earlier, Borth et al. [1] employs Linear SVM for training the ANP detectors. For each of the 1200 ANPs, they employ a one-vs-all SVM classifier. To compare the accuracy of the different classifiers, we train ANP detectors using different kernel settings for SVM and compare each one to see how the different models behave and perform. The different kernels used are:

1. Linear Kernels
2. Polynomial Kernels with degree 1
3. Polynomial Kernels with degree 2
4. RBF kernels
5. Sigmoid Kernels

For this task, we identify 67 ANPs that best capture the different emotions portrayed by the original 1200 long set and train a classifier for each ANP.

4.1.2 Naïve Bayes Binary Classifiers *(Original work)*

In Borth et al. [1], inputs to the Linear SVMs are different image features like colors (RGB), SIFT or GIST, BOW, LBP, Histogram and PHOW (common descriptors for images). Our assumption is that each of these features capture different properties of the image and are inherently independent. Under this assumption, we test a Naïve Bayes classifier for training ANP detectors for images. Using the same set of 67 ANPs, we compare the relative performance of a Naïve Bayes and best SVM classifier. As we discuss in the Experiments section, the Naïve Bayes approach achieves nearly similar accuracy as the best SVM classifier.

4.2 Sentiment Detection Methods

4.2.1 SVM Regression-based Detection
(Re-implementation using SVM instead of Logistic Regression)

We use a sigmoid kernel SVM regression to find the sentiment score of an image. The feature set for this system is the output from the ANP detection phase in the form of a 708 long vector containing the probabilities of that ANP belonging to the image, scaled by the individual sentiment score of the ANP. Borth et al. [1] uses Logistic Regression for this purpose.

4.2.2 Convolution Neural Networks *(Original work)*

The Convolution Neural Networks have been proven to produce very good results in image segmentation and object detection. We attempt to extend the use of CNNs to use it for ANP detection and sentiment label of the image. We use the Deep Learning Toolbox for MATLAB to implement a 6 layered CNN (3 convolution layers and 3 sub sampling layers).

4.3 Sentiment Analysis of Videos *(Original work)*

In this project, we are interested in examining the feasibility and relative accuracy of applying a trained photograph-based image sentiment analysis model such as our own to videos as a method of identifying the graphical features in film that illicit psychophysiological responses so as to classify the expected positive or negative emotional response in humans.

To this end, we use 34 film clips from the FilmStim database [3] as the test set. Each clip in this database is affiliated with an emotion such as sadness, anger, amusement or disgust, which was assigned during an associated study in which emotional responses of humans were recorded during in-lab viewings. In order to prepare the data for efficient and adequate analysis, we have sampled each clip at one frame per second.

The different methods that we have employed to perform sentiment analysis on video are:

1. *Linear SVM* — We parse each frame individually through our pipeline to extract the sentiment score and produce an effective 'mapping of the sentiment' for each of the 34 clips to plot the time variation of the sentiment across the clips. This, as expected, yields a certain degree of mixed results. However, we do find that the model is capable of picking up on changes in trends of similar cinematic compositions. In the end, sentiment scores of each sample snapshot is averaged to provide the final score for the video.

2. *HMM-1* — We believe that the sentiment scores of each frame should have temporal correlation. Usually, an event spans across continuous frames, which should lead to these frames having similar sentiment scores. With this underlying assumption, we use the uncorrelated sentiment score of individual frames to find the hidden correlated sentiment labels of each frame.

3. *HMM-708* — Instead of using the sentiment scores as the observations, we directly take the ANP probability scores as the observations. We assume that the ANP scores follow a Gaussian distribution and use a discrete state HMM with Gaussian observations. The hidden state describes the sentiment label of the frames.

4. *Local Similarity Weighted SVM (LSWS)* — We propose a new method to revise the sentiment scores of the video frames. We revise the sentiment score of each frame obtained from the SVM regression, using the scores of its neighboring frames. These scores are weighted according to a) the cosine similarity and b) the time lag between the frame under reference and its corresponding neighbors. The number of neighboring frames that are taken into consideration while revising the sentiment score of the particular frame is controlled by a factor "τ" called the field width. Time lag refers to the difference between the timestamp of the frames. For instance, if, say, frame number 15 is being processed, its immediate neighbors 14 and 16 will have time difference of 1. Equation 1 depicts how to calculate this score.

$$\forall_i \; \boldsymbol{LSWS(i)} = \frac{\sum_{n=max\left(0,\frac{i-\tau}{2}\right)}^{min\left(\frac{i+\tau}{2},N\right)} \boldsymbol{ANP(I_n)\cdot \cos(I_i,I_n)\cdot TDF(I_n)}}{\sum_{n=max\left(0,\frac{i-\tau}{2}\right)}^{min\left(\frac{i+\tau}{2},N\right)} |\boldsymbol{\cos(I_i,I_n)}|} \tag{1}$$

5 Experiments and Results

5.1 ANP Detection Phase

5.1.1 Comparing Different SVMs

We compare different SVM kernels based upon the accuracy achieved by each on the test set. It is observed that across all ANPs, the sigmoid kernels consistently give the worst performance. The performance of other SVM settings are similar to each other. Figure 2 shows percent accuracies of these different settings. The graph also depicts the best SVM setting selected for each ANP to give the final trained model (red line).

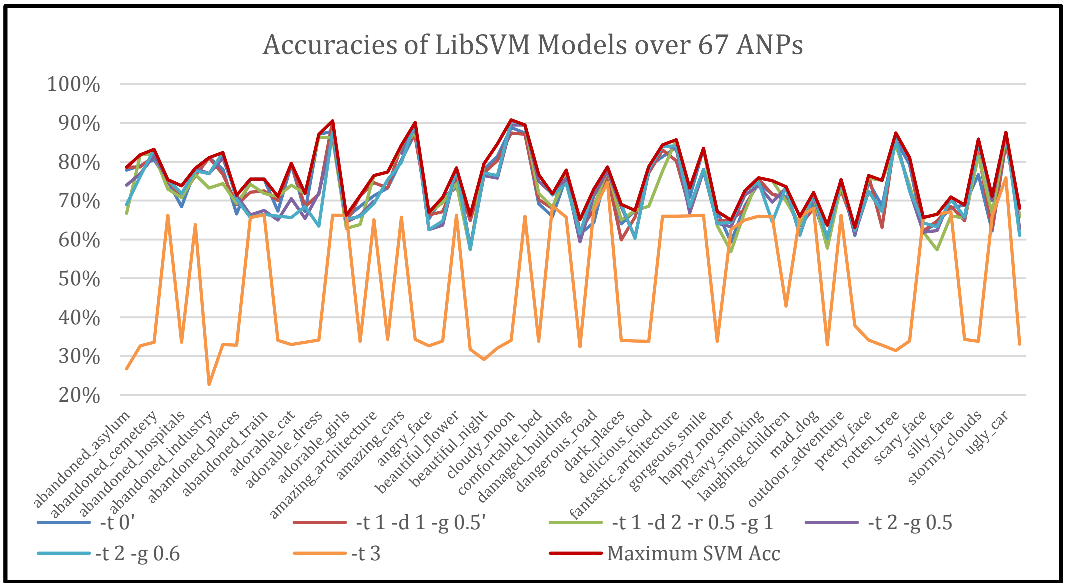

Figure 2: Comparison of 67 ANPs for 6 different kernel settings for SVM classification. The accuracies have been computed by average of runs over 5 different test sets. The best performing kernel for each ANP is shown by the red plot.

5.1.2 Naïve Bayes vs. SVM

We draw a comparison between the binary classifiers for detecting the presence of ANPs in an image trained using Naïve Bayes and the best SVM binary classifier. Figure 3 shows the percent accuracies achieved for all the ANPs. It is observed that although the overall winner is SVM, however, Naïve Bayes classifiers do not lag behind with a huge margin. The difference however, is huge in terms of the time taken to train each classifier. Table 1 shows the average time taken to train a Naïve Bayes and a SVM classifier. Hence, we see that a relatively simpler model (Naïve Bayes) performance is close to the complex SVM model yielding a huge time benefit.

Table 1: Average time taken per ANP to train a binary classifier

AVERAGE TIME TAKEN per ANP (in seconds)	
NAÏVE BAYES	SVM
1.42334	52.35

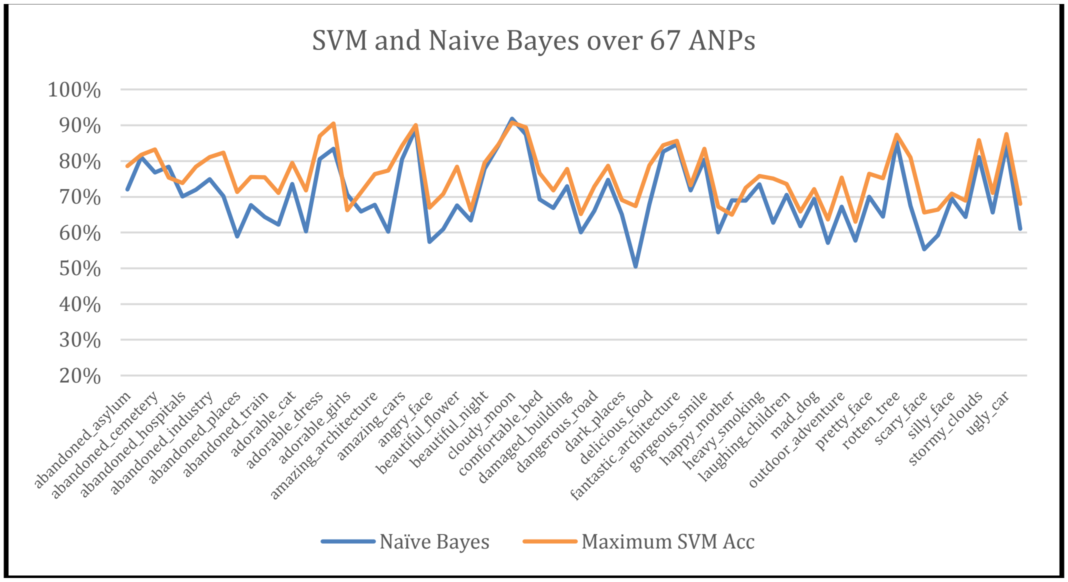

Figure 3: Comparison of accuracies achieved for 67 ANPs from Naïve Bayes and the best SVM trained classifier model. The accuracies have been computed by average of runs over 5 different test sets.

5.2 Sentiment Detection of Images

5.2.1 SVM Regression-based Detection

Experiment: We use multiple SVM settings to find the sentiment of the image. The best performing out of these is Sigmoid Kernels.

Results: The original paper uses Linear SVM (67% accuracy) and Logistic Regression (70% accuracy) to train classifiers for labelling the sentiment (positive or negative) of an image based upon the ANPs that have been detected in the image. Our model is trained using the sigmoid kernel SVM and has achieved 70% accuracy. Table 2 describes the precision and recall achieved.

Table 2: Statistics of the trained model for classifying images as positive or negative sentiment. Our aim is to maximize recall in order to detect as many relevant images as possible.

STATISTICS	VALUES
HOLD OUT CROSS VALIDATION ACCURACY	0.72
ACCURACY	0.70
PRECISION	0.6667
RECALL	0.7143
FSCORE	0.6987

5.2.2 Convolution Neural Networks

Experiment: We test CNNs for detecting the sentiment of an image using a 6 layered deep neural network. Despite having a well-balanced training and test set with equal number of positive and negative examples, the CNN trained models are heavily biased, and always predict the same class.

Analysis: The data set we use for training the CNN models is the set of labeled images from Twitter as provided by [1]. As this is a very small data set (comprising about 1000 images), the resulting train and test set is very limited. Additional fine tuning of the initial parameters is required for CNNs to ensure that they do not get trapped in local minima.

5.3 Sentiment Analysis of Videos

Applying image sentiment detection to the test set of videos has given various results, especially when applying the models that take into account the temporality of frames within the overall clip. Testing results for each video can be viewed interactively using our web interface at:

http://umich.edu/~tzachari/545/#1

Results for different clips may be viewed by switching the url hash value to any number from 1-31, 36, 38, or 61. A snapshot of the interface with comments on usage is shown in Figure 4.

Figure 4: A snapshot of the interactive test-result viewing interface. This particular snapshot depicts the SVM result of the frame, the overall clip score, the top ANP matches for the frame, and the waveform depicting the scores of all the frames in the clip using SVM regression. Source of Snapshot: http://umich.edu/~tzachari/545/#9

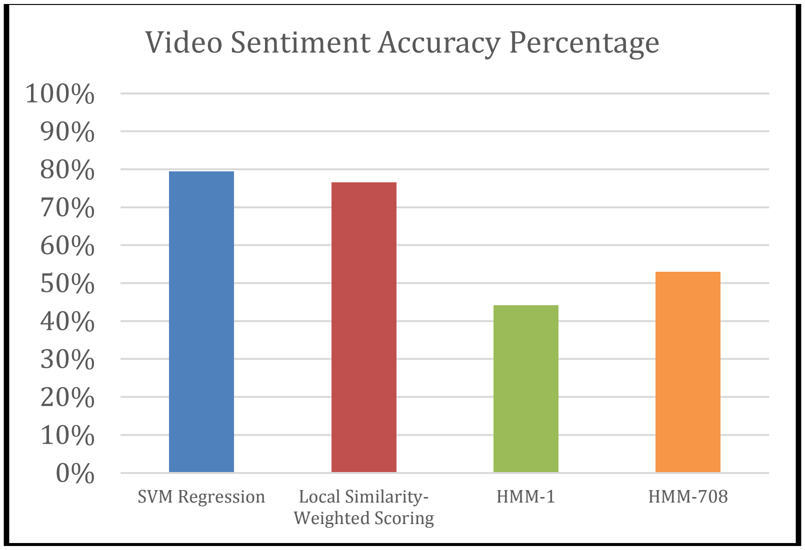

Figure 5: Comparison of accuracies of video sentiment classification over the entire test set achieved using the SVM Regression, LSWS, HMM-1, & HMM-708 techniques discussed.

5.3.1 SVM Regression Based Method

Experiment: We divide the video into frames (sampling one frame per second). Each frame is then passed through the ANP Detectors and Sentiment detectors to get its SVM regression based sentiment score (ranging between -1 to +1). We then take the average of scores of all the frames in a video to arrive at the final sentiment score of the video (ranging from -1, being most negative, to +1 being most positive).

R*esults*:

Figure 4 shows a snapshot of one of the videos and lists the ANPs sorted in order of the probability with which they correspond to the image. It also shows a plot showing the positive and negative regions of the video. We are able to achieve an accuracy of about **80**% using this model as is shown in Figure 5, first bar.

Observations: Largely the sentiments of the frames/videos that are predicted are aligned with the actual sentiments. For scenes in images/frames for which an exact ANP is not present, the most probable ANPs detected very closely capture the sentiment of the original scene. For instance, Figure 4 shows a dying extraterrestrial for which we do not have any ANP. However, the a couple top most ANPs returned are ‘weird bug’ and ‘dying fish’ which seem to be reasonable matches in terms of resemblance and the corresponding sentiment scores, given the limited number of ANPs in the set. This strong detection system leads to good accuracy for our model.

Analysis: Here we analyze the plausible reasons for misclassification of an image’s/video’s sentiment.

1. **Poor performance of some ANPs**: Some of the ANPs are not being correctly identified. Particularly, the ones related to “crying” adjective are misclassified. Our testing till now has helped to identify general subjects and situations the training set seems to lack in representing.

2. **More ANPs Required**: The wide selection of the videos require a wider selection of the ANPs. Some of the ANPs such as those detecting weapons, screaming and explosions are missing from our original selection of ANPs. We need to broaden our ANP base to give true representation of the different types of emotions/objects commonly featuring in the videos.

3. **Lack of Context information**: Sometimes, an image viewed in isolation portrays a

different meaning than when it is part of a complete video. As our method views snapshots in isolation and does not have any information about the context of the video, it results in labelling positive images as negative or vice versa. For instance, one of the videos in our data set shows scenes from a dry comedy in which most of the individual frames are wrongly labeled as negative.

5.3.2 HMM Results

Experiment: We use the Gaussian HMM implementation of python-sklearn to learn a first order HMM with discrete hidden states (possible values: +1, -1). The implementation we call HMM-1 uses one-dimensional observations (the uncorrelated SVM regression sentiment scores of the frame) and implementation HMM-708 uses the multi-dimensional observations (probability scores of each of the 708 ANPs for the frame obtained from the ANP Detection phase). For both implementations, expectation maximization is executed for approximately 100 iterations and then Viterbi algorithm is applied to find the best possible state sequence. We run different trials for the HMMs and picks the model corresponding to the maximum log probability score. This initializes the training system with random values and hence ensures that we are not actually selecting a local minima.

Observations: Contrary to our initial expectations, the HMMs have performed poorer than SVM, achieving only about **44**% and **50**% accuracy (Figure 5 third and fourth bar respectively). HMM-708 performed slightly better than HMM-1. This is as expected and thereby corroborates the correlation between the visual concepts (ANPs) and the sentiment of the frame.

Analysis:

1. The distribution of the ANP scores, as depicted in Figure 6, does not quite resemble a Gaussian distribution and hence, may be one of the reasons for poor performance of the model.
2. Another reason for the poor performance is that we directly use the ANP probabilities as observations. However, this will lead to all ANPs having the same weightage towards the final score. For instance, an ANP – ‘Happy Cloud’ should have lesser weight than ANP ‘Destructive Weapon’. In the absence of differential weightings, our model cannot identify strong biases.

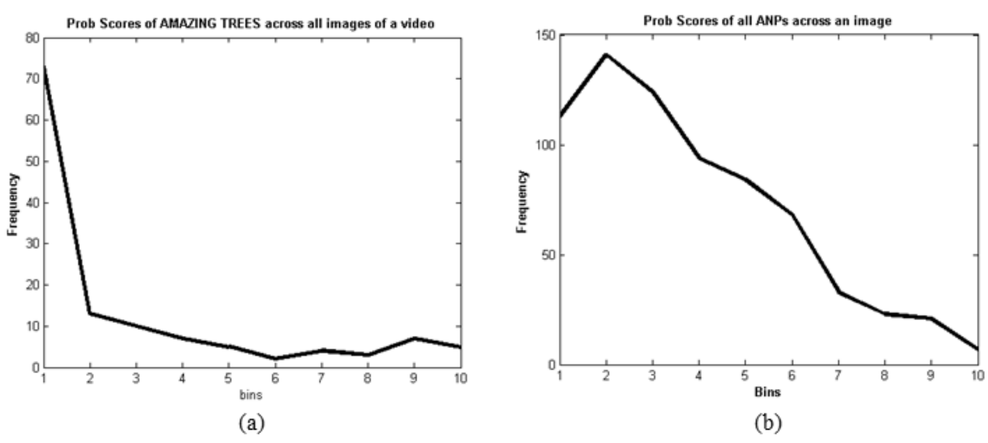

Figure 6: Histograms of probability scores for: a) ANP ‘Amazing Tree’ in FilmStim Video #1, and b) all ANPs in a frame of FilmStim Video #1

5.3.3 Local Similarity-Weighted SVM Results

Experiment 1: Calculating the revised scores of the video frames

We use the formula from Equation 1 to revise the regression based sentiment scores of each frame of each video. These scores are then smoothened and then averaged to give the final sentiment score of these videos. We have experimented with different values of field width, or number of neighboring frames considered (ranging from 2 to 128) and the following three different types of temporal decay factors: **-first**: Default value 1, **-second**: Exponential: exp(-timeLag) and **–third**: Linear: (1/timeLag). In this paper, we report the results using the default constant value. The model achieves an accuracy of about 77% on the test set using a window size of 8 (Figure 5 second bar).

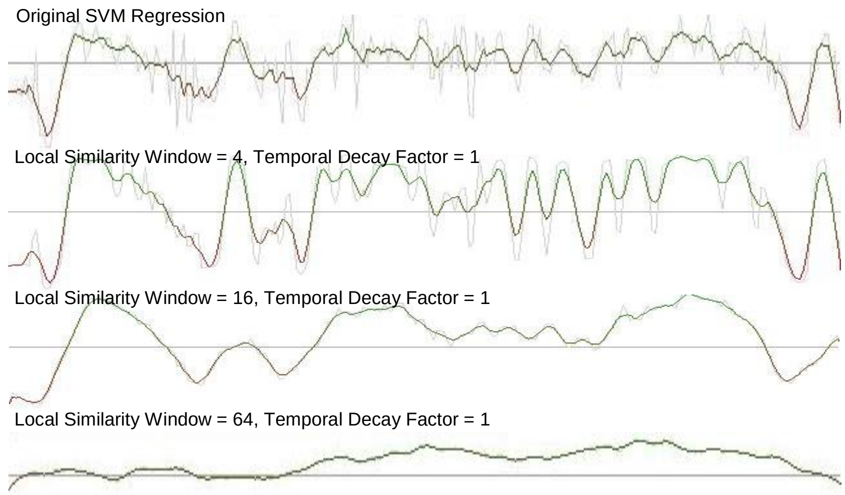

Figure 7: Plots of the sentiment Scores of FilmStim Video #31, using Local Similarity-Weighting with various field widths (local similarity windows) and default temporal factor of 1.

Observations:

1. One of the most interesting observations is that the revised scores are generally more confident than the original SVM regression in predicting true sentiment label of a particular frame. A frame that is previously correctly labeled, sees a greater tendency towards the score. A frame that is previously incorrectly labelled as negative or positive is often moved in the direction of the correct label.
2. With increasing field width (τ), the sentiment curve smoothens as shown in Figure 7.
3. The linear temporal decay factor smoothed the sentiment plot across frames for any video. The exponential and the default decay factors captured the variations in the sentiment better. The various shapes of the sentiment plots are shown in Figure 8.
4. In a couple of clips, though the LSWS score seems appropriate for a particular section of frames, it does not reflect the overall sentiment of the clip and the weighting is too biased. This accounts for the slightly poorer performance than original SVM. We believe fine tuning of the parameters can fix issues such as these.

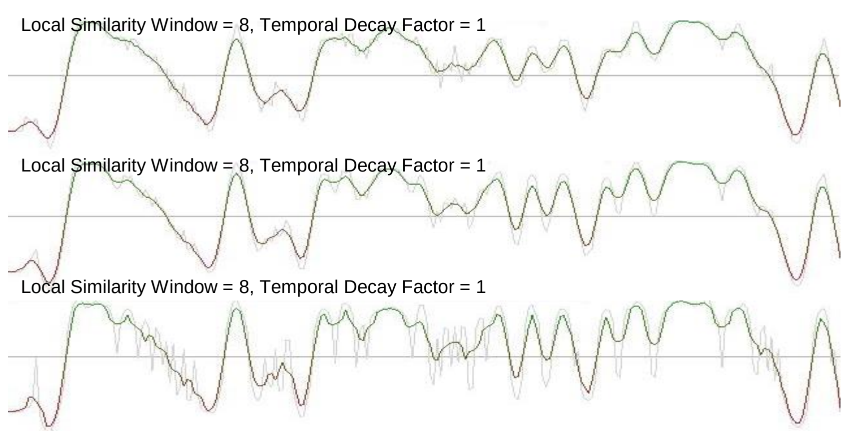

Figure 8: Plots of the sentiment scores of FilmStim Video #31, using Local Similarity-Weighting with various temporal factors and similarity window of 8.

Experiment 2: Shuffling the video sequence.
In order to check the effect of temporal alignment of the frames on the scores, we shuffle the sequence of the frames and then recalculate their sentiment scores using SVM regression and the LSWS method.

Observations: Figure 9 shows the plot of the sentiment scores returned from the shuffled sequence. For easy comparison, the frames have been stitched back in original sequence. It can be noted that the SVM regression provides relatively neutral results for a large segment of frames, where LSWS method shows more confidence in the sentiment conveyed due to weighting according to the cosine similarity of the frame with its 'neighborhood' of frames. To verify that consideration of the neighborhood is indeed the cause of the result, we shuffle the frames of the clip (effectively changing the neighborhood sets for each frame) and after applying LSWS, find that doing so leads to very different contributions to the weighting, and, therefore, different cosine similarity scores for any given frame.

Figure 9: The plots of the scores of LSWS (Weighted), Shuffled LSWS (Shuffle), and original SVM (SVM), for FilmStim Video #36. Shapshot source: http://umich.edu/~tzachari/545/#36.

6 Conclusions and Future Work

In this project, we have presented that a frame by frame, image-based sentiment analysis of a video is a simple yet very good indicator of the overall sentiment of the video yielding a high accuracy. This technique makes it possible to analyze any type of videos with no restriction on their lengths. Our new method LSWS gives better results when we look at the frames individually as compared to the kernel SVM regression, but, in our test results, the latter has shown slightly better accuracy in terms of the overall video sentiment. We have presented one way of applying the HMMs to our problem statement and as shown, they perform better when they have knowledge of all the ANPs.

In the future, we hope to improve upon our model to better detect more complex features, such as facial expressions. Additionally we plan on fine tuning the parameters for our local similarity-weighted scoring and attempt to better implement HMM and CNN. Finally, we intend on further exploring the various applications in which our model and results can be utilized.

Acknowledgments

We would first like to thank Professor Satinder Baveja for his help and motivation in pursuing this project. Additionally, we would like to acknowledge Borth et al. [1], Schaefer et al. [3], and Carvalho et al. [4], all of whose studies and corresponding datasets have played a large part in our project. We have used MATLAB and Python for all programming that necessary for this project.

References

[1] D. Borth, R. Ji, T. Chen, T. Breuel and S.-F. Chang, "Large-scale Visual Sentiment Ontology and Detectors Using Adjective Noun Pairs," in *ACM Int. Conference on Multimedia (ACM MM)*, Barcelona, Spain, 2013.

[2] S. Siersdorfer and J. Hare, "Analyzing and Predicting Sentiment of Images on the Social Web," *ACM*, 2010.

[3] A. Schaefer, F. Nils, X. Sanchez and P. Philippot, "Assessing the Effectiveness of a Large Database of Emotion-eliciting Films: A New Tool for Emotion Researchers," *Cognition & Emotion*, vol. 24, no. 7, pp. 1153-1172, 2010.

[4] S. Carvalho, J. Leite, S. Galdo-Álvarez and Ó. F. Gonçalves, "The Emotional Movie Database (EMDB): A Self-Report and Psychophysiological Study," *Applied Psychophysiology and Biofeedback*, vol. 37, no. 4, pp. 279-94, 2012.

[5] J. A. Bilmes, "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," 1998.

[6] B. C. Lovell, "Hidden Markov Models for Spatio-Temporal Pattern Recognition and Image Segmentation," in *International Conference on Advances in Pattern Recognition*, Kolkatta, 2003.

[7] D. Paul, "Speech Recognition Using Hidden Markov Models," *Lincoln Laboratory Journal* , vol. 3, no. 1, 1990.

[8] A. Krizhevsky, I. Sutskever and G. Hilton, "ImageNet Classification with Deep Convolution Neural Networks," *NIPS*, pp. 1106-1114, 2012.

[9] "Flickr API," [Online]. Available: http://www.flickr.com/services/api.

[10] "Visual Sentiment Ontology," [Online]. Available: http://visual-sentiment-ontology.appspot.com.

[11] Scikit-Learn. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.hmm.GaussianHMM.html.

[12] "LibSVM," [Online]. Available: www.csie.ntu.edu.tw/~cjlin/libsvm/.