Machine Learning Models for Political Video Advertisement Classification

Boudhayan Banerjee
Dept. of Computer Science
Iowa State University
Ames, United States
bbanerji@iastate.edu

***Abstract*—Investment in online political ad marketing is gaining traction very rapidly. In United States, the 2016 presidential election campaign witnessed substantial increase in political advertisement expenditure in online platforms like YouTube. Therefore, political researchers are interested in analyzing trends of political ads in online medium. But currently there is no existing method or application that can classify political advertisement from a large dataset of online ads. In this paper, we attempted to solve this problem by proposing a model which can automatically classify political video advertisements using machine learning algorithms such as Support Vector Machine, Linear Regression and Naïve Bayes classifier. We will also focus on feature engineering for this classification problem. We applied text features and non-text features like color and facial features for classification purpose. We trained 3 different models with different feature set and compare results among them. We also created an ensemble with these 3 models and achieved an F1-score of 0.97.**

***Index Terms*—political advertisement classification, machine learning, feature generation, OCR, text classification.**

I. INTRODUCTION

Video is one of the major platform for marketing. According to data published by YouTube in 2017, 1 billion hours of videos are viewed daily by 1 billion users. The increase in video consumption has led increase in investment in video advertising. According to the data published by Google [47] since April 2015, people have watched more than 110 million hours of political candidates and issues related content on YouTube, which is 100 times more than all the contents aired on CNN, C-Span, MSNBC and Fox News combined. Consequently, political candidates are investing more campaign resources in the digital advertising platform, to reach out to the viewers. In the 2016 general US presidential election, the political campaign ad spending was $9.8 billion according eMarketer [48]. In 2012 presidential election campaign $8.81 billion was spend according to same source. But the more interesting fact is that, US digital ad spending, increased substantially from $159.2 million in 2012 to $1076.7 million in 2016. Therefore, Digital ad platform is the fastest growing ad delivery platform for political campaigns.

The collection and analysis of political videos ads are thus becoming an integral and critical part of research for political science researchers. In order to understand more about the political advertising, political science researchers need a platform that can be used to explore and analyze the content of online political video advertisements. During 2016 presidential election campaign we discovered nearly ~ 8000 ads from YouTube. According to our discoveries, we found that political advertisements are only 3-4% of the total video advertisements appearing on YouTube. Our goal is to identify the political ads from rest of online video ads. But at present there no automated way to identify political advertisements. Manual identification of such advertisements is expensive and time-consuming process, as one needs to watch hundreds of hours of videos in order to classify political ads. Therefore, we need a better method for political ad identification. We used machine learning models to classify political advertisements, as many researchers had great success using machine learning models in various classification and regression problems.

In this paper, we introduce a classification model that can classify political campaign ads from other online video advertisements. We used a dataset of ~ 1700 videos to train our model. For

classification of political advertisements, we investigated and reviewed different potential features. These features include both text based features and non-text based or content based features. Text based features include text captured from each frame of the video, where a single frame represents one second of a video. Text based features also includes the audio transcription of the video. To extract the text from the video frames we reviewed few OCR techniques. We investigated and compared the OCR results between Google Vision API [26] and Tesseract OCR [50], two of the most commonly used OCR applications.

We also investigated several non-text based features including color features and facial features. Yang Song et al., [1] in their paper described the importance of color histograms in video classification. The idea behind this approach is US political advertisement often features more red and blue colors. We tried to leverage that characteristics and used it as feature. We calculated the percentage of red and blue pixels from last 3 seconds of the advertisements. We found these values are statistically different in political and non-political ads using Hypothesis Test. Accordingly, the color features e.g., percentage of red and blue pixels are considered to be used as features in the classification problem.

Another non-text based feature we investigated is to count the average number of faces appearing in each frame of the advertisements. This can potentially be an important characteristic in order to classify political advertisements, because political advertisements tend to feature large gathering of people in campaign events. Hence, we calculated the average number of faces in each frame of the advertisements in our dataset. We also explored the distribution of the values of facial features in political and non-political ads. We found the median value of facial feature in political advertisement to be higher than non-political advertisements. To ensure the difference is statistically significant we performed Hypothesis Test on our data. The p-value from the result of the Hypothesis test small enough to select this as a feature for the classification problem. We will discuss in detail about the feature extraction process in section 3.

Third type of feature we considered, is the presence of certain keywords in the advertisement. According to the Federal Election Committee guidelines every political campaign ads must include certain disclaimers [41]. These disclaimers usually include the information regarding the sponsorship of the advertisement and also if the message conveyed in the ad is authorized by the candidate or not. We will mention these disclaimers as Keyword in later discussions. We used existence of such keyword in the video as a feature. We will discuss the rationale behind the consideration of all these features in section 3.

We trained three different linear models. The first model featured only the text features. The second model comprised of all the non-textual features like percentage of red and blue pixels in last 3 seconds of the video and average number of human faces in each frame. The third model used the existence of keyword in the advertisement as feature. In our training, model 1 achieves an F1 score of 0.96, whereas model 2 and 3 achieves F1-score 0.94 and 0.72 respectively. We went one step further and created an ensemble with these 3 models. We obtained an F1-score of 0.97 with our ensemble model.

Rest of the paper is organized as follows. In section 2 we will discuss about the related work in this field of study. Section 3 describes our proposed work. Section 4 presents the experiments and the results. In section 5 we will discuss the future scope of our proposed work. Section 6 summarizes the conclusion.

II. Related work

The idea of using text based features from the individual frames of a video along with other non-text based features is most closely related to work by Yang Song et al., [1] where the feature extraction process was divided into two sub-categories of features, e.g., Text based feature extraction and content based feature extraction. Text based features included video title, description, search queries and

keywords. Whereas content based features included color histogram, edge features, Histogram of textons, face features, Color motion and shot boundary features and audio features. We are using similar approach for combining text based features and non-textual features for our classification model.

Previously the idea of skimming video to engineer features from image frames and audio has been introduced by Smith et al [49]. In their paper, they described video characterization technique using image and language understanding. They performed language analysis on the transcript to identify important audio regions and used TF-IDF (Term frequency Inverse document frequency) to measure the relative importance of words for the video document. To detect video frames, they performed scene segmentation by measuring comparative color histogram differences. They detected significant changes in the weighted color histogram of successive frames. Significant objects e.g., human faces and texts were also identified from the video frames. We followed similar approach to separate the image frames and audios from the source video to extract texts appearing in each individual image frames as well as to extract transcription of the audio. We also performed human face detection on the video frames.

There is an automated coding of political video advertisements [43] which is able to classify types of political ads as attack ads, promotional ads and contrast ads. This approach uses rule based methods as well as text classification using extracted texts from video frames and audio transcript. They also performed classification of advertisements based on sentiment analysis of the image frames. But this method does not identify political ads out of large dataset of online advertisements.

Thorsten Joachims introduced the use of Support Vector Machines for learning text classification from examples [6]. This method operates on the idea of representing a text document as a feature vector of words. To reduce the large size of the feature vectors stopwords are removed from the text documents. Also, only the words that appear in the text document at least 3 times are considered as a feature. This paper provides evidence towards the use of SVM as a text classification tool. Because SVMs use overfitting protection scheme which is independent of number of features, SVMs perform well in high dimensional input space. Also, the document vectors are sparse. i.e. only few entries in the vector contains non-zero elements and SVMs are well suited for sparse vectors.

Grant Schindler et al. describes the “bag of words” model to represent a video using different combinations of spatial and temporal descriptors [44]. We will use “Bag of words” representation in our text based feature model classification.

C.D. Paice introduced an evaluation method for stemming algorithms [8]. Stemming algorithms reduces morphological variants of words in its stem. This paper also describes the concept of stemming errors and their implication on variants of stemmer. A light stemming algorithm tries to avoid over stemming errors. But consequently, introduces under stemming errors. Whereas heavy stemming algorithms performs more aggressively to remove all sort of endings from the words and as a result introduces over stemming errors.

Sami Abu-El-Haija et al. used the technique of decoding videos at one frame per second [2] to extract image frames of a video. We will use this process of decoding to convert video advertisements into image frames.

Viola et al. [40] introduces a new machine learning approach which is known as Viola-Jones object detection framework to detect objects from images rapidly with high detection rates. They use Haar-like features, which is a scalar product between the input image and some Haar like template. We have used the scheme proposed in this paper to detect different faces appearing in each individual frame of the ad videos. We will discuss more about Viola-Jones object detection framework in section 3

III. PROPOSED WORK

We prepare a balanced dataset of ~ 1700 advertisement videos for our training dataset. We maintain the balance in the dataset in order to reduce the number of misclassification due to under-representation of one class in the dataset. We collect the political advertisements from two different sources [31,32]. We assemble non-political advertisements dataset from set of ~ 8000 ads from YouTube which we discovered during 2016 presidential election campaign.

The complete process is divided into 3 sub-processes, e.g., Data Pre-processing, Feature Extraction and Classifier Training. Fig 1 shows the overview of the entire sequence of process. Once the training dataset is prepared, we preprocess the dataset to extract features from the training data.

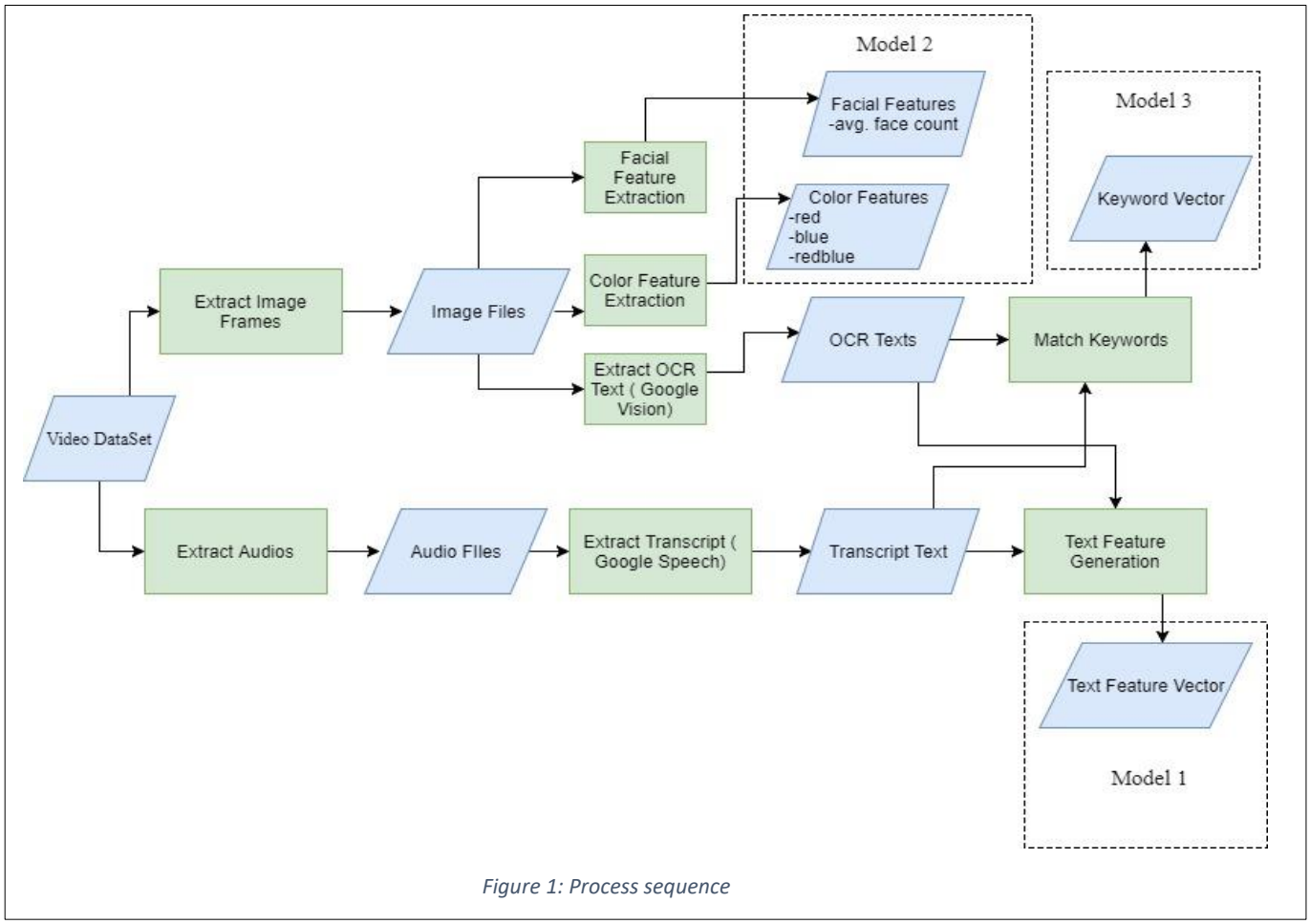

Figure 1: Process sequence

A. Data Pre-processing:

We segment each individual ad videos into image frames, with one frame per second of video. E.g. a 30 second ad video will produce 30 frames. S. Abu-El-Haija et al. explains this video decoding scheme of extracting 1 frame from each second of a video [2] in their youtube-8m paper. Next, we segregate audio from each advertisement video in the training dataset.

We use optical character recognition to detect texts appearing in each frame of all the ad videos in our training dataset. We investigated OCR programs like Tesseract OCR [50], an open source OCR application and Google's cloud vision API [26]. The challenge of using an OCR application in complex images is that, each image needs to be preprocessed differently in order to obtain optimal result from the OCR application. The preprocessing procedures include rescaling of the image, i.e., each image should have a DPI of at least 300. Also, images need to be de-skewed and absent of any kind of noise that arises due to random variation of brightness. We compared the performance between Tesseract OCR and Google Vision API. We performed test on both the OCR applications on a set of 20 images, where all the images were set to 300 DPI with 1300 x 768 resolution. We manually listed out words appearing in each of these 20 images and tested each image against two OCR applications. We measured the accuracy of the OCR applications by computing the ratio of number of correctly identified word and number of actual words identified by human. TABLE 1 shows the comparative result of Tesseract OCR and Google Vision API on the test dataset of 20 images.

TABLE 1. Comparison between Google Vision and Tesseract OCR

	Google Vision	**Tesseract OCR**
Accuracy Measure (number of correctly identified word / number of actual words identified by human)	0.734	0.213

We can improve tesseract OCR's performance by tuning individual input images. But the process is challenging and time consuming to pre-process each individual frame in a large dataset in order achieve better OCR performance.

Therefore, we applied Google's cloud vision API to detect the texts from the image frames. Although cloud vision API works better or as good as any other OCR applications, it still sometimes misclassifies a text or falsely identify something as text. For example, in Figure some of the texts are classified wrong.

Figure 2: Google OCR result

In order to control this behavior of the OCR application we checked whether the identified text is a valid English word. We performed data cleaning on the text documents produced by OCR application.

Data Cleaning: We are only interested in the valid English words that are identified by OCR application. To validate each word, we check them against a dictionary. But our text documents consist of a number of named entities e.g., name of candidates and politicians, name of states and other places, different current political issues etc. A standard dictionary usually does not include most of the named entities. On that account, we opt for Wikipedia to validate words identified by OCR application. Wikipedia is a free online encyclopedia. Furthermore, it is the most comprehensive general reference framework in the internet. Wikipedia also provides open API to check if an input word has a valid reference in Wikipedia. We use this API to check each individual word identified by OCR application [46]. We store the final extracted text in separate files for all the advertisement videos in the training dataset.

We transcribe the extracted audio files using Google's Speech API [27]. This API although not perfect, it only generates valid English dictionary words as outputs. Therefore, we do not need to clean the output texts. After the completion of the process the transcription text for each individual advertisement videos are stored in separate files.

B. Feature Extraction:

We use a classification model with both textual and non-textual features. First, we describe the process of textual feature generation.

1. *Text Based Features:*

We already extracted texts from individual frames of the advertising videos. We also collected the transcription of each advertising videos. Therefore, each advertising videos in the training dataset has corresponding OCR and transcription text documents respectively.

In the next step, we removed the stopwords and most of the punctuations from each text documents. Stop words are the most commonly used words in a language that carries little to no significance in classification. Stopwords has little semantic weight, hence not used in any information retrieval purpose. Natural Language Processing library NLTK provides a list of most commonly used English stop word list. In our application, we use this Stop word list to reduce the size of text feature vector space and to eliminate less significant features from the feature set.

Afterwards, we vectorize the training text dataset. Vectorization is a process of converting a text document dataset into feature vectors. We followed a vectorization approach which is commonly known as *Bag of words* or "Bag of n-grams" representation [45]. N-gram indicates a sequence of n continuous elements. In our approach, we used 'word' as unit for n-gram model. We explored with different combination of n-gram range to obtain the best model for our classification purpose. TABLE 2 explains the range of n-gram representation where a 'gram' represents a 'word'.

TABLE 2. n-gram representation of tokens

n-gram range	**features**
(1,1)	paid, for, by, candidate
(1,2)	paid, for, by, candidate, paid for, for by, by candidate
(1,3)	paid, for, by, candidate, paid for, for by, by candidate, paid for by, for by candidate

The vectorization process can be sub-divided into following sub-processes. First, we tokenize each individual OCR and transcript text document. Tokenization is a process of separating tokens from a body of text by using whitespaces and punctuations. The words that are at least two letters long are considered as tokens. After tokenization, we will assign a tf-idf weight against all the tokens in each individual OCR and transcription text document. Tf or term-frequency of a token in a document, is the number of times the token is present in a text document. IDF or inverse document frequency is a logarithmic function of ratio of number of text documents in the corpus and number of documents where the token appears. TF or term frequency score of each token is calculated by,

$$TF(t, d) = 1 + \log f_{t,d}, \text{ or zero if } f_{t,d} \text{ is zero}$$

where t = term, d = document, $f_{t,d}$ = count of term t in document d

IDF or Inverse document frequency is calculated as,

$$IDF(t,D) = \log (N / |\{d \epsilon D : t \epsilon d\}|$$

where N = total number of documents, D = set of all documents

TF-IDF is simply product of TF and IDF.

Vectorization process is complete when all the n-grams in the training corpus dataset is assigned a tf-idf weight. Each individual token in the n-gram model is considered as a feature in the classification model.

2. *Non-Text Based Features:*
   We investigated through the collected dataset to find any special characteristics of the dataset that can serve as a feature in classification. We explored political advertisement dataset to observe that political videos in general has more red and blue colors. It is due to the fact that, red and blue represents the color of republican and democrat party respectively. Also, American flag sometimes appears during the ad. The candidate's logo is also shown near the end of the video, which in most of the cases has red or blue or both colored theme associated with it.

   We took this idea and measured the most significant color in each political ad and non-political ad. But the result was not significant, because political ad frames have high concentration of red and blue color only towards the end of the video, not during the whole length of the video. Therefore, we considered only last 10% length of video to measure most significant color. But this time also we couldn't find any evidence towards our initial hypothesis.

   We then considered the percentage of red and blue pixels instead of finding whether red and blue are the most significant color in a frame. Figure shows that although red and blue are not the most significant color in the given frame, it can still be considered as good feature if we consider the percentage of red and blue in the video frame. The process of measuring each pixel in an image within a large dataset is time consuming process. We observed most political videos are between the length of 15 seconds to 60 seconds. Therefore, on average we were processing last 2 to 6 seconds of video in most of the cases. We optimized the process a little bit by processing the last 3 seconds of each video. Which turns out to produce more significant difference of feature values between political and nonpolitical ads. From figure 3 it can be observed that, for this particular frame most significant color is some variation of black and grey. Despite the fact, red and blue are not most significant color in this frame the percentage of red and blue pixels is significant and can potentially identify political advertisements.

Figure 3: image has higher percentage of red and blue pixels although most significant color is different variation of black

Consequently, we considered the percentage of red pixel, percentage of blue pixel and percentage of combination of both red and blue pixel in the last 3 seconds of the advertisement as potential features.

Similarly, we observed that more number of people appears in political advertisements. The advertisement campaign often represents various political conventions, debates, town hall meetings, house parties and other gatherings of group of people. Therefore, average number of human faces appearing in a political video is potentially higher than a non-political one. Accordingly, we considered average number of human faces in a frame as a potential feature for the classification problem. But a potential feature cannot be considered as a true feature in classification until we can determine there is a correlation between the political advertisements and the afore mentioned features. Next, we will discuss the process of extracting color and facial features from the training dataset.

a. Color Features:

To identify whether a pixel in an image is red or blue we measure the HSV value of that pixel. We choose HSV value to represent a pixel over other color models e.g., RGB because Hue, Saturation and Value of a color can be expressed as a range and we need to calculate whether HSV value of the pixels in video frames falls within a certain range. Approximate HSV color range for red is *h = (0-10), s = (100 - 255), v = (100 - 255)* and is *h = (170-180), s = (100 - 255), v = (100 - 255).* Approximate HSV color range for blue is *h = (110-130), s = (50 - 255), v = (50 - 255)* [40].

For each frame of an advertising video we observe the HSV value of all the pixels. If the HSV value of a certain pixel falls within the HSV range of red color, it is considered as a red pixel and if it falls within the HSV range of blue color, it is considered as blue pixel. For each advertisement in the training dataset, we calculate the percentage of red and blue pixels in last 3 frames. The percentage of red, blue and combination of red and blue pixels in last 3 seconds of an advertisement will be referred to as red, blue, redblue respectively. Fig 4,5 shows the percentage of red and blue pixels calculated from last 3 seconds of a political advertisement and a non-political advertisement respectively.

Although Figure 4,5 shows example of a single political and non-political ad, we can have some notion about how color features will differ in political and non-political advertisement. To validate the statistically significant difference between two classes, we will perform Hypothesis Test which is discussed later in the section.

% Red pixel :1.346, % blue pixel: 19.473

Figure 4: Percentage of red and blue color in last 3 seconds of a political advertisement

% Red pixel :4.286, % blue pixel: 0

Figure 5: Percentage of red and blue color in last 3 seconds of a nonpolitical advertisement

b. Facial Features:

Human faces appearing in each frame of the advertisement were detected by Haar Cascade algorithm [38], It is a machine learning based approach, where a cascade function is trained on a dataset containing images with faces and images without faces.

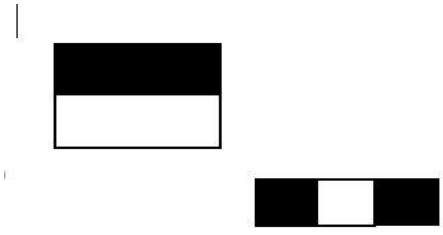

Figure 7: Rectangle Haar Features

Figure 6: Haar features applied on an Image

The general idea behind Haar Cascade algorithm is, human face region can be divided into light and dark spaces. E.g., the eye region tends to be darker than cheek region. Also, eyes are darker than the nose bridge. The lighter and darker regions are represented using white and black rectangle respectively. From figure 6 and 7 we can observe how a rectangular Haar feature is applied on a sample image to identify human face. Each feature is a single value obtained by subtracting pixel value of light region from pixel value of dark region. These features are called rectangle features and the value is calculated as,

$$Value = \Sigma \text{ (pixels in black area)} - \Sigma \text{ (pixels in white area)}$$

We observe the average number of faces in each frame of the advertisement. The average number of faces in each frame of an advertisement will be referred to as face. Fig 8 shows the process of detecting number of human faces in each frame of advertising videos.

Number of Face Detected: 45

Figure 8: Number of Face Detected

Hypothesis Testing: Once the feature vectors are generated for each advertisement in the training dataset, we observe the distribution of the data that belong to both political and non-political classes by plotting box plot of each features. The mean, median and standard deviation of facial and color features are given in Table3.

TABLE 3. STATISTIC OF FEATURES

Feature	**mean**		**median**		**Std. deviation**	
	political	***nonpolitical***	***political***	***nonpolitical***	***political***	***nonpolitical***
red	5.6	5.84	3.22	1.57	9.19	13.65
blue	4.1	6.73	0.83	0.6	10.95	15.38
redblue	9.7	12.57	5.7	4.56	13.97	19.63
face	1.81	0.76	1.36	0.62	1.94	0.59

We can observe from Table3 the mean percentage of red, blue and combination of red and blue pixels in last 3 seconds is higher in non-political advertisements. This happens due to more number of outliers' present in non-political advertisements. Fig 7,8 and 10 shows the distribution of color features as box

plots for red, blue and redblue feature respectively. We can observe from these box plots that there are large number of outliers. Therefore, mean value of political and non-political features are almost same, although median values are higher in political ads. To make a decision whether to include these feature in classification algorithm we need to perform Hypothesis test on these features.

Similarly, Fig 9 shows more number of outliers in political advertisements for the face feature. We observe one of the outlier to be 31.i.e., the average number of faces appearing in each frame of that advertisement is 30.38. Upon further investigation we observe, the advertisement shows a gathering of large number of people. Hence, the Haar Cascade Algorithm [38] detects almost 30-31 human faces on average for each frame of that ad. Fig 7 shows the individual frames of the video.

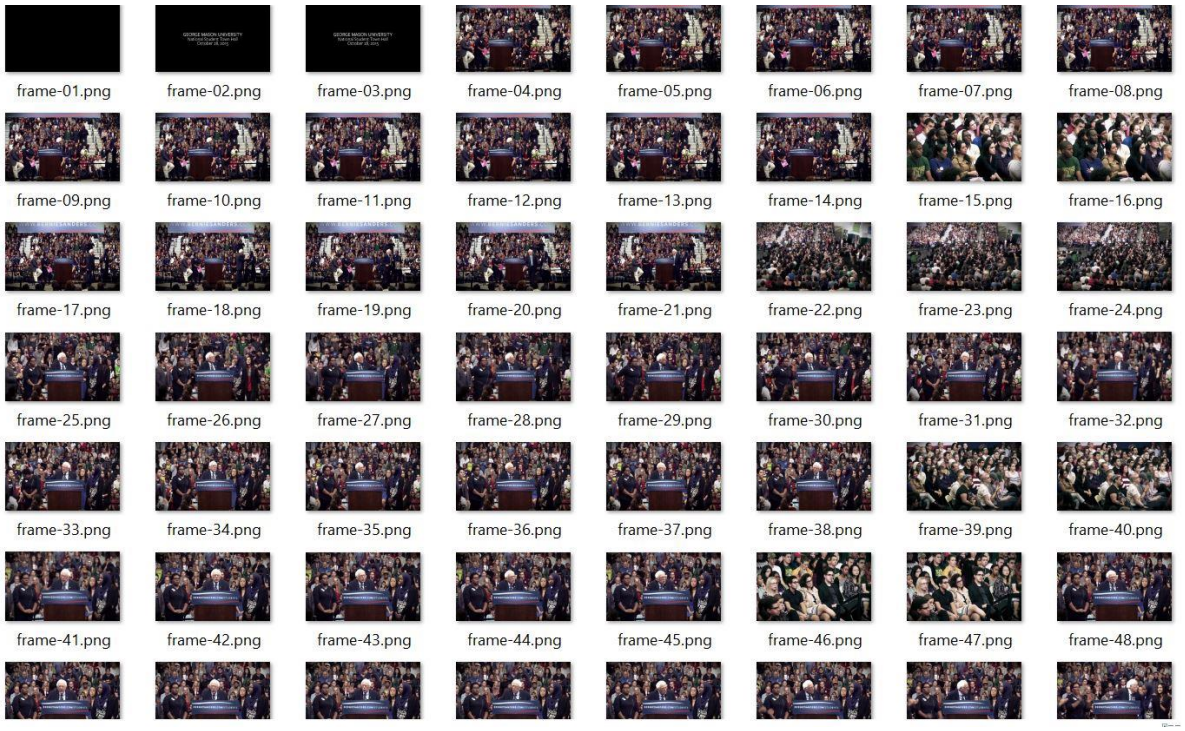

Figure 9: Frames of a political ad where facial feature is an outlier

Although Box and whisker plots provides good representation of the distribution of the data, to find the statistical significance of the features we perform t-test on facial and color features. Our null hypothesis is that, there is no significant difference of facial and color feature value of a political advertisement and a non-political advertisement. If the p-value is less than 0.05 then we can discard our null hypothesis. Otherwise we accept the null hypothesis and discard that potential feature. We define our null hypothesis and alternative hypothesis as following,

H_0: $red_{political} = red_{non\text{-}political}$

H_a: $red_{political} > red_{non\text{-}political}$

H_0: $blue_{political} = blue_{non\text{-}political}$

H_a: $blue_{political} > blue_{non\text{-}political}$

H_0: $redblue_{political} = redblue_{non\text{-}political}$

H_a: $redblue_{political} > redblue_{non\text{-}political}$

H_0: $face_{political} = face_{non\text{-}political}$

H_a: $face_{political} > face_{non\text{-}political}$

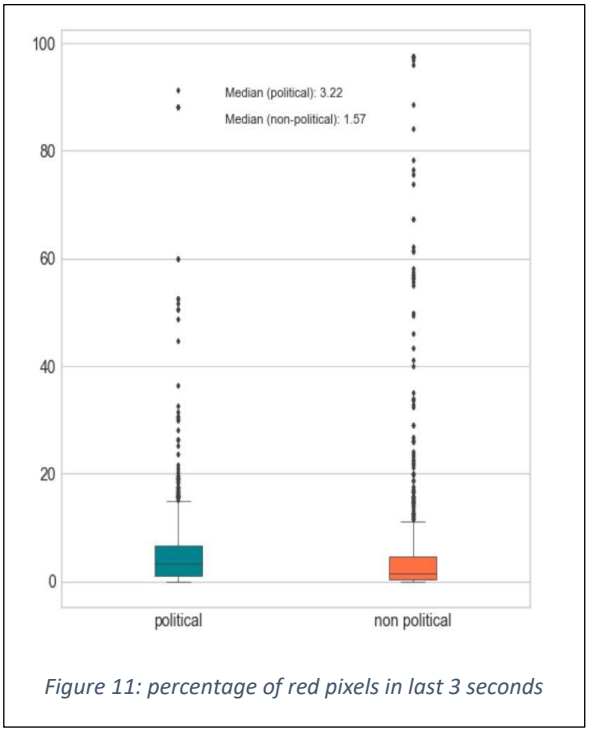

Figure 11: percentage of red pixels in last 3 seconds

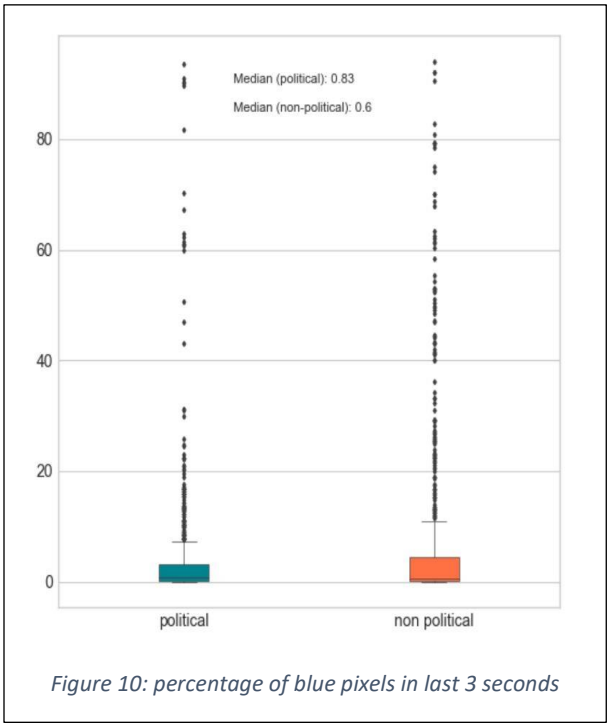

Figure 10: percentage of blue pixels in last 3 seconds

Along with t-test we also perform Bartlett [36] and Levene [37] test to measure the statistical difference between the feature values of political advertisements and non-political advertisements. There are different tests we can perform for Hypothesis test. We choose to perform Bartlett and Levene test to detect variability or spread of our data, as these are identified as common method of quantitative techniques by National Institute of Standards and Technology [51]. The results are given in Table 4.

TABLE 4. RESULTS OF HYPOTHESIS TEST

Feature	t-test	Bartlett	Levene
red	0.67	6.63e-30	0.08
blue	5.03e-05	9.54e-23	2.09e-05
redblue	0.0005	1.01e-22	1.59e-06
face	1.18e-46	4.09e-218	3.48e-15

From the Hypothesis test result we can reject null hypothesis for blue, redblue and facial features, as the p-value given by all 3 tests are smaller than threshold p-value or significance level of 0.05. But in

case of red feature p-value is more than 0.05 for both t-test and Levene test. Therefore, we cannot reject the null hypothesis. But we also observed that p-value can be affected because the presence of a large number of outliers. Therefore, we considered against discarding red as feature and included it in our feature set.

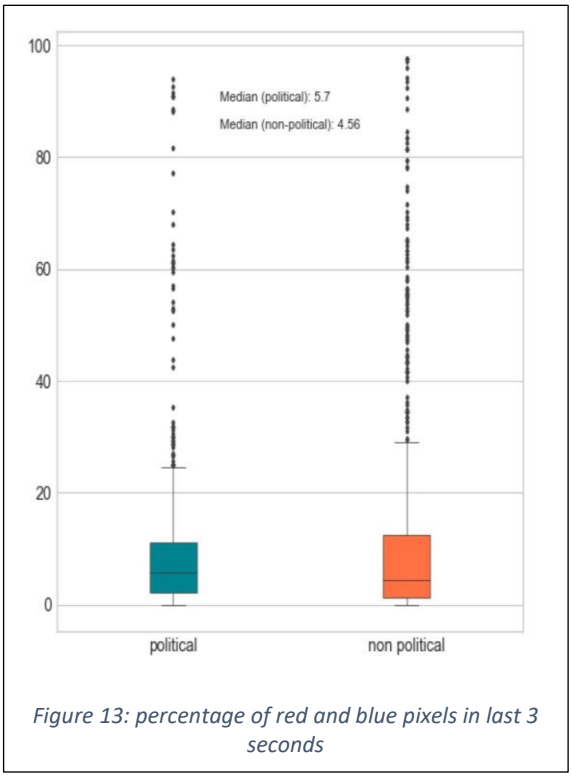

Figure 13: percentage of red and blue pixels in last 3 seconds

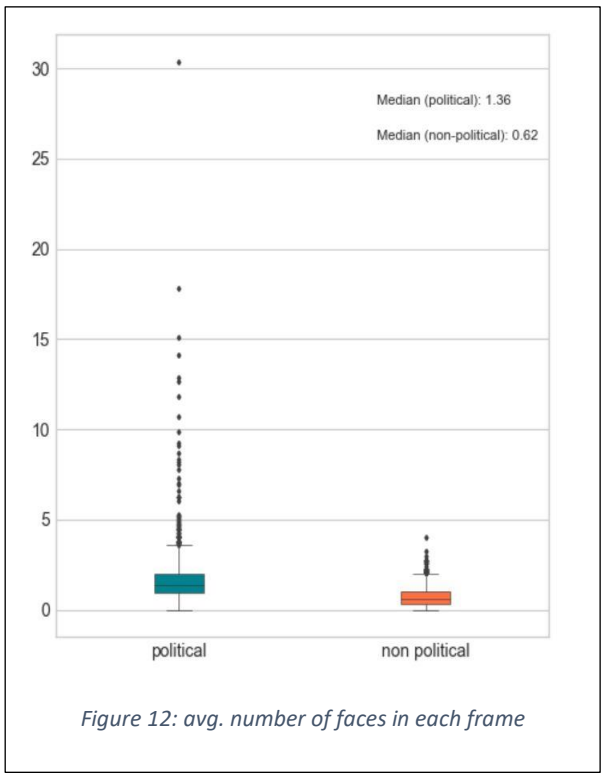

Figure 12: avg. number of faces in each frame

3. *Keyword Features:*

According to Federal Election Commissions guidelines it is a requirement that political advertisements include certain disclaimers [39]. These disclaimers state the information about the sponsors of the advertisements. Disclaimer also specifies whether the intended communication is authorized by any candidate or candidate's committee. These disclaimers will be referred to as keywords. Locating the keywords in an advertisement will almost certainly classify the advertisement as political. Although there are following caveats for this approach.

- Keyword feature depends on the performance of Google Vision API and google Speech API. When both the APIs fail to locate keywords from the video frames or from the audio, keyword only feature fails to correctly identify advertisement. Google Vision API fails often when video is low resolution or disclaimer are shown using smaller or fancier fonts. Google Speech API can fail when the disclaimer words are spoken too fast.
- Non-political advertisement sometime includes certain keywords that are used as political campaign ad disclaimer.
- Keyword matching does not work if the advertisement is in a different language. e.g., Spanish.

Therefore, we will use the keyword feature as one of the feature in our classifier. It is a binary feature. If the advertisement contains the keyword the value of the feature is 1, otherwise it is 0. We inspect the text documents created by OCR application and the audio transcription of each advertisements in the training dataset to retrieve this Keywords. If we obtain any of the Keyword from a certain advertisement, we assign a feature value 'true' against that ad, we assign 'false' otherwise. The list of Keywords we use in our experiment is given in Table 5.

TABLE 5. DISCLAIMER KEYWORDS

Keyword
Paid for
Approved by
For president
Authorized by
Responsible for
Approve this message
candidate
president

C. Classifier Training

We separate the text features, non-text features and keyword features and use 3 separate models for classification. We separate the features because, we wanted to investigate the effect of different features separately. Also, text features are sparse and have high dimensionality which can affect non-text features and keyword features in process.

We train 3 models on the training dataset. For the classification algorithm, we compared Naïve Bayes classifier, Logistic Regression classifier and Support Vector Machine with linear kernel. The rationale behind choosing a linear model instead of a nonlinear is, on a relatively small dataset nonlinear model can cause overfitting. In section 4 we will describe the performance of each classifier on the training dataset. We tune the hyperparameters of the classification algorithm with Grid search algorithm. Grid search algorithm performs an exhaustive searching through a given subset of hyperparameter space. In support vector machine, regularization constant C is tuned using Grid search algorithm.

Although C is a continuous parameter, for grid search we select a reasonable finite set of values C = [1,10,100,1000]. Grid search algorithm compares the result of each of these values and selects the best value for the hyperparameter.

IV. EXPERIMENTS AND RESULTS

A. Preparing Training Dataset

We prepared a dataset consisting of 1682 advertisements, with 841 advertisements each for political and non-political advertisements. The political advertisements were collected from Stanford Political Communication Lab [31] and Political TV ad archive [32]. The non-political advertisement dataset was curated from a dataset of ~ 7000 advertisement []. We ensured that the non-political advertisement dataset has participation from different genres. E.g. the non-political dataset consists of ads of car, food & drinks, banks & other financial institution, movie & tv-show trailers, insurance, health products, electronics, travel, pet care, personal care etc.

We use 5-fold cross validation to train our dataset. 5-fold cross validation shuffles and divides the entire training dataset into 5 random parts and then holds out one part for validation and trains the classifier on the remaining 4 parts for one iteration. In the next iteration, another part will be selected for hold-out and classifier will be trained on remaining 4 parts. This process will continue 5 times for 5-fold cross validation. Cross validation score is given as the mean of the results of 5 iterations. We use cross validation in order to reduce the chance of overfitting the data.

B. Performance of Classifiers

We use Scikit-learn [33], a machine learning library in Python which provides various tools for data mining and data analysis. We use 3 linear classifier algorithm Naïve Bayes, Logistic Regression and Support Vector Machine with linear kernel in this paper.

We train our classifiers on 3 different models. The first model comprises of only the text features from OCR and audio transcripts. Second model consists of non-text based features which includes red, blue, redblue and faces. Finally, the third model consists of only the keyword feature. For simplicity, we will mention these models as model 1, model 2, model 3 respectively. We use Grid search algorithm to fine tune the values of hyperparameters. For model 1 we experimented with n-gram ranges (1,1), (1,2), (1,3) and (1,4). We achieved optimal result with n-gram range of (1-3).

We use performance metrics such as precision, recall, F1-score and cross validation score to measure the accuracy of our classification model. The performance metric F1-score is calculated as a weighted average of precision and recall. Precision is defined as,

Precision = Number of ads correctly classified as political / (Number of ads correctly classified as political + Number of ads incorrectly classified as political)

Whereas Recall is defined as,

Recall = Number of ads correctly classified as political / (Number of ads correctly classified as political + Number of ads incorrectly classified as non-political)

F1-score is calculated as,

F1-score = 2 x (precision x recall) / (precision + recall)

The results of the model fitting scores are shown in Table 6.

TABLE 6. PERFORMANCE OF 3 CLASSIFICATION MODELS

Methods	Model 1				Model 2				Model 3			
	p	r	f	cv	p	r	f	cv	p	r	f	cv
NB	0.94	0.93	0.93	0.86	0.51	0.48	0.49	0.53	0.95	0.95	0.95	0.96
SVM	0.96	0.97	0.96	0.96	0.75	0.74	0.74	0.73	0.94	0.94	0.94	0.95
LREG	0.94	0.96	0.95	0.96	0.69	0.55	0.48	0.62	0.96	0.96	0.96	0.85

Note: p = precision, r = recall, f = F1-score, cv = cross-validation score

From Table 5 we can find that Model 1 performs better overall with Support Vector Machine and Linear Regression. With both of these classifiers we are able to get a cross-validation score of 0.96. F1 Score is slightly higher at 0.96 in SVM compared to 0.95 in linear regression model. For model 2, SVM performs better than other algorithms but overall F1-score and cross validation score is lower compared to Model 1. Model 3 performs similar with all 3 classification algorithms achieving highest F1 score of 0.96 with Linear regression model and 0.96 cross validation score with Naïve Bayes model. Naïve Bayes model performs rather poorly compared to other two classification algorithms because Naïve Bayes does not consider interaction among features, it treats them as independent. Although Naïve Bayes performs in similar way in Model 3, because Model 3 contains just a single feature and there is no interdependency among features.

C. Testing unknown Dataset

Prior to train the classification algorithms on our training dataset we prepared another dataset that remain unknown to the machine models. After the completion of training process, we validate the performance of the models on this unknown dataset. For classifying an advertisement, we run the classifier on our pre-trained models and collect the probability scores from all 3 models. Then we compare the probability score of 3 models and select the one with highest probability. This method is called ensemble model. This is closely related to voting classifier [52]. There are other ensembling methods. But ensembling algorithm generally uses same features but different classifier algorithm. Our approach uses different set of features in 3 different models. Therefore, we decided to follow the approach of comparing probability score of participating models. We created a holdout set of 18 videos which are not seen by any of our trained models. We ran our pretrained models on these set. Table 7 shows the result of the experiment.

TABLE 7: PERFORMANCE ON UNKNOWN DATASET

	Model 1	MODEL 2	MODEL 3	P1	P2	P3	ENSEMBLE MODEL	GROUND TRUTH
	non_political	non_political	political	0.71	0.78	**0.98**	political	political
	non_political	non_political	political	**0.99**	0.79	0.98	non_political	non_political
	political	political	political	**1**	0.82	0.98	political	political
	political	non_political	political	**1**	0.53	0.98	political	political
	political	non_political	non_political	**1**	0.51	0.92	political	political
	political	political	non_political	**1**	0.52	0.92	political	political

	political	political	non_political	**1**	0.99	0.92	political	political
	political	political	non_political	**1**	0.5	0.92	political	political
	political	non_political	non_political	**1**	0.7	0.92	political	political
	political	non_political	political	**1**	0.83	0.98	political	political
	political	non_political	political	0.97	0.58	**0.98**	political	political
	non_political	political	political	0.85	1	**0.98**	political	political
	non_political	non_political	political	**0.99**	0.7	0.98	non_political	non_political
	political	political	political	0.99	**1**	0.98	political	political
	non_political	non_political	political	0.68	0.52	**0.98**	political	political
	non_political	political	political	0.62	0.56	**0.98**	political	non_political
	non_political	political	political	0.69	0.57	**0.98**	political	political
	political	political	political	0.92	0.72	**0.98**	political	political
Precision	**100%**	**88%**	**77%**				**94%**	
Recall	**78%**	**53%**	**66%**				**100%**	
F1-score	**0.87**	**0.66**	**0.71**				**0.97**	

We can observe that in Model 1 we get a precision score of 100%, although the recall score is 78%. Model 2 has the lowest recall score of 53%. Model 3 has precision score of 77% and recall score of 66%. Our ensemble model has precision score of 94% and recall score of 100%. Thus, the ensemble model has the highest F1-score among all the models with 0.97.

V. Conclusions

Political candidates are investing increasing amount of resources in Online Advertisement platform. Therefore, to understand more about the political advertisement political researchers need a platform to investigate and explore the content of online political advertisements. In this paper, we made an attempt to provide a method which can classify political advertisements from a large dataset of online ads. We considered three different models with different feature sets. These features are combination of both text based and non-text based features. We analyzed each of the non-text based feature to measure the level of significance. i.e., if the value of the particular feature in a political ad is significantly different from that of a non-political ad, so that they can be distinguished based on that feature.

We found the text based model which comprises of the text features extracted from the image frames by OCR application and the audio transcripts perform with a F1-score of 0.96 and cross validation score of 0.96 with Support Vector Machine as classifier. Model 3 which comprises of the keyword feature attains a F1-score of 0.96 with Linear regression and cross validation score of 0.96 with Naïve Bayes classifier. Model 2 which comprises of non-text based features e.g., color and facial features obtains a maximum F1-score of 0.74 and cross validation score 0.73 with Support Vector Machine.

We experimented our pre-trained model on a hold-out set of 18 videos. We compared the probability score of these 3 models and choose a class based on highest probability score. We achieved a F1-score of 0.97 with the ensemble model.

REFERENCES

[1] Song, Y., Zhao, M., Yagnik, J., Wu, X.: Taxonomic classification for web-based videos. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 871–878 (2010) J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. arXiv preprint arXiv:1609.08675, 2016K. Elissa, "Title of paper if known," unpublished.

[3] C. Apte, F. Damerau, and S. Weiss. Text mining with decision rules and decision trees. In Proceedings of the Conference on Automated Learning and Discovery, Workshop 6: Learning from Text and the Web, 1998.

[4] Paice, C.D. An Evaluation Method for Stemming Algorithms. In: ACM SIGIR Conference on Research and Development in Information Retrieval, 1994, pp. 42-50.

[5] K. Nigam, A.K. McCallum, S. Thrun, and T. Mitchell, "Text Classification from Labeled and Unlabeled Documents Using EM," Machine Learning, vol. 39, nos. 2/3, pp. 103-134, 2000.

[6] T. Joachims, "Transductive Inference for Text Classification Using Support Vector Machines," Proc. 16th Int'l Conf. Machine Learning, pp. 825-830, 1999.

[7] I. Dhillon, S. Mallela, and R. Kumar. A divisive information-theoretic feature clustering algorithm for text classification. JMLR, 3:1265–1287, 2003.

[8] Paice, C.D. (1994). An evaluation method for stemming algorithms. In W.B. Croft & C.J.van Rijsbergen (Eds.), Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (pp. 42–50). London: Springer-Verlag.

[9] Paice, C.D. (1990). Another stemmer. *SIGIR Forum*, **24**, 56–61.

[10] Lovins, J. B. Development of a stemming algorithm. Mechanical Translation and Computational Linguistics, 11, pp. 22-31, 1968.

[11] Goldsmith, J., Higgins, D., and Soglasnova, S. Automatic language-specific stemming in information retrieval. In Cross-language information retrieval and evaluation: Proceedings of the CLEF 2000 workshop, C. Peters, Ed.: Springer Verlag, pp. 273-283, 2001.

[12] Porter, M. F. An algorithm for suffix stripping. Program, 14 (3), pp. 130-137, 1980.

[13] Popovic, M. and Willett, P. The effectiveness of stemming for natural-language access to Slovene textual data. JASIS, 43 (5), pp. 384-390, 1992.

[14] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," in Lecture Notes in Artificial Intelligence 5782, W. Buntine, M. Grobelnik, and J. Shawe-Taylor, Eds. Berlin, Germany: Springer, 2009, pp. 254–269.

[15] C. Apte, F. Damerau, and S. Weiss. Text mining with decision rules and decision trees. In Proceedings of the Conference on Automated Learning and Discorery, Workshop 6: Learning from Text and the Web, 1998.

[16] C. Apte, F. Damerau, and S. Weiss. Towards language independent automated learning of text categorization models. In Proceedings of the 17th Annual ACM/SIGIR conference, 1994.

[17] Thorsten Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In European Conference on Machine Learning (ECML), 1998.

[18] Y. Yang. An evaluation of statistical approaches to text categorization. Journal of Information Retrieval (to appear), 1999.

[19] Y. Yang. Sampling strategies and learning efficiency in text categorization. In AAAI Spring Symposium on Machine Learning in Information Access, pages 88{95, 1996.

[20] Y. Yang and J.P. Pedersen. Feature selection in statistical learning of text categorization. In the Fourteenth International Conference on Machine Learning, pages 412{420, 1997.

[21] A. Dasgupta, P. Drineas, B. Harb, V. Josifovski, M.W. Mahnoney, Feature selection methods for text classification, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, CA, USA, 2007, pp. 230–239.

[22] J. Neumann, C. Schnorr, G. Steidl, Combined SVM-based feature selection and classification, Machine Learning 61 (2005) 129–150.

[23] Z. Zhao, H. Liu, Spectral feature selection for supervised and unsupervised learning, in: Proceedings of the 24th international Conference on Machine Learning, Corvalis, Oregon, 2007, pp. 1151–1157.

[24] University of Wisconsin, Department of Political Science. (1998). The University of Wisconsin Advertising Project. Available: http://wiscadproject.wisc.edu/project.php

[25] E. Hersch, Hacking the Electorate: How Campaigns Perceive Voters: Cambridge University Press, 2015.

[26] Google Cloud Vision API. [Online]. Available: https://cloud.google.com/vision

[27] Google Cloud Speech API. [Online]. Available: https://cloud.google.com/speech/

[28] C. Sujatha and U. Mudenagudi, "A Study on Keyframe Extraction Methods for Video Summary," Computational Intelligence and Communication Networks (CICN), 2011 Int'l Conf. on, Gwalior, 2011, pp. 73-77.

[29] H.-Y. Liu and H. Zhang, "A sports video browsing and retrieval system based on multimodal analysis: SportsBR," in Machine Learning and Cybernetics, 2005, pp. 5077-5081.

[30] A. Jungherr, "Twitter in Politics: A Comprehensive Literature Review," SSRN, vol. 2402443, 2014.

[31] Political Communication Lab.[Online]. Available: http://pcl.stanford.edu/

[32] Political TV Ad Archive. [Online]. Available: https://politicaladarchive.org/

[33] Scikit-Learn. [Online]. Available: http://scikit-learn.org

[34] Hsu, Chih-Wei, Chih-Chung Chang, and Chih-Jen Lin. "A practical guide to support vector classification." (2003): 1-16.

[35] substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[36] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[37] Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel et al. "Scikit-learn: Machine learning in Python." Journal of Machine Learning Research 12, no. Oct (2011): 2825-2830.

[38] Snedecor, George W. and Cochran, William G. (1989), Statistical Methods, Eighth Edition, Iowa State University Press.

[39] Levene, H. (1960). In Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling, I. Olkin et al. eds., Stanford University Press, pp. 278-292.

[40] Viola and Jones, "Rapid object detection using a boosted cascade of simple features", Computer Vision and Pattern Recognition, 2001.

[41] Federal Election Commission. [Online]. Available: https://www.fec.gov/

[42] OpenCV Python. [Online]. Available: http://opencv-python-tutroals.readthedocs.io

[43] L. Qi, C. Zhang, A. Sukul, W. Tavanapong and D. A. M. Peterson, "Automated Coding of Political Video Ads for Political Science Research," *2016 IEEE International Symposium on Multimedia (ISM)*, San Jose, CA, 2016, pp. 7-13.

[44] G. Schindler, L. Zitnick and M. Brown, "Internet video category recognition," *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Anchorage, AK, 2008, pp. 1-7.

[45] Harris, Zellig. Distributional structure. Word, 1954.

[46] Wikipedia API. [Online]. Available: http://en.wikipedia.org/w/api.php

[47] YouTube Data. [Online]. Available: https://www.thinkwithgoogle.com/marketing-resources/content-marketing/political-ads-video-content-influence-voter-opinion/

[48] "Digital Political Ad Spending to Skyrocket in 2016". [Online] Available: https://www.emarketer.com/Article/Digital-Political-Ad-Spending-Skyrocket-2016/1013861

[49] M. A. Smith and T. Kanade. Video skimming and characterization through the combination of image and language understanding. In CVPR, 1997

[50] Tesseract OCR. [Online]. Available: https://github.com/tesseract-ocr

[51] National Institute of Standards and Technology. [Online]. Available: http://www.itl.nist.gov/div898/handbook/eda/section3/eda35.htm

[52] D. Ruta, B. Gabrys Classifier selection for majority voting Information Fusion, 6 (1) (2005), pp. 63-81.