Network-Centric Computing (NCC) Special Issue

Name-It: Naming and Detecting Faces in News Video

Shin'ichi Satoh [1]
National Center for Science Information Systems (NACSIS)

Toshio Sato
School of Computer Science, Carnegie Mellon University

Michael A. Smith
School of Computer Science, Carnegie Mellon University

Yuichi Nakamura [2]
Institute of Information Sciences and Electronics, University of Tsukuba

Takeo Kanade
School of Computer Science, Carnegie Mellon University

Running head: Name-It: Naming and Detecting Faces in News Video

Contact: Shin'ichi Satoh,
R&D Dept., NACSIS,
3-29-1 Otsuka, Bunkyo-ku, Tokyo 112, Japan.
ph: +81-3-3942-6956
fax: +81-3-5395-7064
email: satoh@rd.nacsis.ac.jp

[1] The author had been a visiting scientist at CMU from April 1995 to April 1997.
[2] The author had been a visiting scientist at CMU from March 1996 to December 1996.

Abstract

We have developed Name-It, a system that associates faces and names in news videos. The system is given news videos, which include image sequences and transcripts obtained from audio tracks or closed caption texts. The system can then either infer possible name candidates for a given face, or locate a face in news videos by name. To accomplish this task, the system takes a multi-modal video analysis approach: face sequence extraction/identification from videos, name extraction from transcripts, and video caption recognition. Each method includes several advanced image and natural language processing techniques: face tracking, face identification, intelligent name extraction using dictionary, thesaurus, and parser, text region detection, image enhancement, character recognition, and the integration of these techniques. The success of our experiments demonstrates the benefits of a multi-modal approach to video analysis.

List of Symbols

$\in, <, >, \geq, \leq$

$x_f, y_f, x_r, y_r, x_l, y_l$

w_f, Fr

d_f, S_f, σ_f

F_i, F_j

$I_e, I_h, I_i, I_{i+1}, \cdots, I_{i+n}$

x_0, y_0

$\mathcal{N}$

$||(x, y)||$

$\lfloor \frac{x}{4} \rfloor, \lceil \frac{x}{4} \rceil$

I_n, I_r

m_c

$x(i)_j^s, p_i^s$

$x(i..)$

$d_c, \hat{d}_c, c_c, \varepsilon$

C^*

N_a, N_b

F_p, F_q

$t_{F_i}, t_{start,F_i}, t_{end,F_i}, t_F$

$o_f, O_f(t; F), \sigma_t$

$N_j, t_{N_j,k}, s_{N_j,k}$

O_n, δ, S_t

$\infty, \otimes, \Delta$

$C_i, t_{C_i}, t_{start,C_i}, t_{end,C_i}, t_C, S_t'$

S_c, θ_c, w_c

1 Introduction

In recent years there has been an increased demand for multimedia applications, including: video on demand, digital libraries, video editing/authoring, etc. Current multimedia data consists of vast amounts of image, video, audio, and text information, into which a modicum of essential "content" has been absorbed. In achieving multimedia applications with video, it is crucial to develop methods to access and extract the contents of video data. Video data is multi-modal, i.e., it may include image sequences, audio (including speech), closed-captions, transcripts, etc. Therefore, a multi-modal approach is an effective means to obtain the contents of video data. Vision/image processing and natural language processing should play an important role.

We propose Name-It [1, 2], a system that associates names and faces in news videos, as a part of the Informedia project [3] at Carnegie Mellon University. It's basic function is to guess "which face corresponds to which name" in news videos. In other words, Name-It detects faces with corresponding names in news videos as content information. Main contributions of Name-It are:

- Revealing effectiveness of multi-modal video analysis.
- Demonstrating importance of extracting real "content" information (face-name association).

We take a multi-modal approach to accomplish this task; i.e., Name-It makes use of several information sources available from news videos: image sequences, transcripts, and video captions. Name-It detects face sequences from image sequences, and extracts names from transcripts. Transcripts can be obtained from audio tracks using a proper speech recognition technique with an allowance of recognition errors. On the other hand, most of news broadcasts in U.S. have closed-captions; and it is the worldwide trend that broadcasts are going to have closed-captions in the near future. Therefore, we use closed-caption texts as transcripts for news videos. We use CNN Headline News as our primary target. Video captions are text information superimposed on frames of video, thus they are obtained from image sequences. They are directly attached to frames of videos, and represent explanatory descriptions of videos. Alternatively, transcripts do not necessarily give explanations of videos, even though corresponding transcripts and image sequences are correlated in time. Given those information sources, Name-It associates extracted faces with extracted names using the correlation of their timing information. Video captions are also taken into account as supplementary information. To associate faces and names, Name-It integrates several advanced image processing/natural language processing techniques: face sequence extraction/identification from videos, name extraction from transcripts, and video caption recognition. The accuracy of each technology is not always high, but by integrating these results, the system will acquire better results.

By providing face-name association, Name-It performs “individual detection” rather than mere “face detection,” because associated faces and names correspond to a certain individual. Particularly, the individual is a person of interest in news video topics. As a consequence, Name-It enables several potential applications, including: (See Figure 1.)

- News video viewer which can interactively provide personal description of the displayed face,
- News text browser which can provide facial information of names,
- Automated video annotation generation for faces.

After introducing related research in Section 2, we describe an overview of Name-It in Section 3. Each elementary technique is then described: face sequence extraction/identification in Section 4, name extraction in Section 5, and video caption recognition in Section 6. Then, there is the association method of analysis results in Section 7, experimental results in Section 8, and the conclusion of the paper in Section 9.

2 Related Work

Face identification is one of the key technologies which support Name-It. There are image database systems which can perform face similarity matching; they include MIT Photobook [4] and the Virage system. These two systems use the eigenvector-based method for face similarity matching [5]. It is noteworthy that Photobook was applied to more than 7,500 face images of about 3,000 people to obtain successful results. The results reveal that eigenvector-based face similarity matching works well to some extent.

Video analysis is one of the most popular research topics today. Among researches in this area, video parsing [6, 7] is more related to our approach. In their approach, once a news video is given as a target, the video is decomposed into segments or shots, then these shots are classified based on the structure of the video. In doing this, several techniques, i.e., cut detection, color histogram calculation and comparison, camera motion analysis, motion segmentation, etc., are employed. Although videos can be "parsed," i.e., the structure of videos can be analyzed, these methods do not provide "content" information such as object identification or topic classification in news videos.

Piction system [8] identifies faces within given captioned photos, typically in newspapers. The system extracts faces from a photo and analyzes captions to obtain geometric constraints among faces which will appear in the photo, then labels each face with each name. The system works fine due to the fact that captions give direct explanatory descriptions of photos. Actually, the system requires the captions to have geometric descriptions for faces, e.g., "top row, from left, are Michael, Brian, ..." On the other hand, as for news videos which are the target of Name-It, even though they also give both text information (transcripts) and visual information, transcripts do not necessarily give explanation of videos. Instead, transcripts provide the main stories, while visual information is used as supplementary information.

3 Overview of Name-It

Typical news video is composed of several topics, and each topic has a corresponding person(s) of interest. Figure 2 shows the typical structure of a news topic. In this topic, U.S. president Bill Clinton is the person of interest. We set the primary goal of Name-It to associate faces and names of persons of interest in news video topics.

We employ the architecture shown in Figure 3 to achieve this goal. Name-It must extract faces from image sequences and names from transcripts, both of which correspond to persons of interest. However, these tasks are hard to accomplish. Faces of persons of interest tend to appear under several conditions, e.g., frontal, close-up, long duration, etc. But faces which meet these conditions do not always correspond to persons of interest; i.e., there is no perfect method to extract faces of persons of interest only by image sequence analysis. Meanwhile, to extract names of persons of interest, in-depth semantic analysis of the transcript is necessary. This is not achieved, even though selecting names in text is achieved with sufficient accuracy [9]. Therefore, we extract faces and names which *are likely to* correspond to persons of interest. The system employs face detection and tracking to extract face sequences, and natural language processing techniques using a dictionary, thesaurus, and parser to locate names in transcripts (See Figure 3.).

Given extracted faces and names, Name-It is requested to associate a corresponding face and name. Since transcripts do not necessarily give explanation of videos, there is no straightforward method to associate faces in videos and names in transcripts. However, by observing the typical news video composition given in Figure 2, we can assume that a corresponding face and name are likely to coincide. Namely, a face-name pair which coincides might be an associated face-name pair. However, there are potential difficulties to associate face and name: Due to lack of necessary faces or names, and possible multiple correspondence of faces and names. For example, even a face of a person of interest in a topic is successfully extracted, it may be the case that no correct name is found which coincides with the face. As an example of multiple correspondence, assume that topic-A is about Clinton and Dole, and topic-B is about Clinton and Gingrich. The system cannot decide whether a face (of Clinton) shown in topic-A corresponds to name “Clinton” or “Dole,” or whether a face (of Clinton) shown in topic-B corresponds to name “Clinton” or “Gingrich.” To compensate for this drawback, Name-It gives priority to a face-name pair which coincide *in more topics* and outputs the pair as an associated face-name pair. Obviously, face similarity (or identification) is required to evaluate face-name association, e.g., to identify the faces in topic-A and topic-B. Thus the system regards these faces identical, and can infer the face coincides in more topics with name “Clinton” than with others (“Dole” or “Gingrich”). By using face identification, the problem of lack of faces or names is also resolved; even a face does not coincide with the correct name, other faces identical to this face are expected to coincide with the correct name.

In addition, video caption recognition is also employed to obtain face-name association. Video captions are superimposed texts on frames of a video, and therefore represent literal information. They are directly attached to image sequences, and give an explanation of the video. In many cases, they are attached to faces, and usually represent a person’s name. Thus video caption recognition provides rich information for face-name association. However, video captions do not necessarily appear for all faces of person of interest. Instead, they are used as a supplement to transcripts: a face, shown like a person of interest, but not mentioned in transcripts, is often given video caption of the name. To achieve video caption recognition, text detection and character recognition techniques are employed (See Figure 3.).

Finally, results obtained by these techniques should be integrated to provide face-name association. We use a co-occurrence factor, which represents a likelihood factor that a face and a name correspond to each other, as a unified measurement integrating multi-modal analysis. This integration should be done so that integration results give better face-name association results, even given results of analysis are imperfect. As shown in Figure 3, to compensate for the problems of lack of faces or names and multiple correspondence of faces and names, face similarity is employed to evaluate the co-occurrence factor. Since character recognition for video captions cannot be perfect due to the poor quality of video images, it is compensated for by an inexact string match method. As a result, although each analysis may not discriminate faces (or names) of persons of interest in topics, association results may eventually correspond to face-name pairs of persons of interest in topics.

4 Face Information Extraction

This section describes extraction of faces which might correspond to persons of interest in topics. We employ face detection and tracking to detect face sequences in videos. Face similarity is then evaluated using an eigenface-based method. To enhance face similarity evaluation, the most frontal view among faces of a detected face sequence is selected and used for the eigenface method. Finally, given videos as input, the system outputs a two-tuple list: timing information (start $\sim$ end frame), and face identification information.

4.1 Face Tracking

Face tracking consists of 3 components; face detection, skin color model extraction, and skin color region tracking (See Figure 4.). The following sub-sections describe the face tracking components.

4.1.1 Face Detection

First, Name-It applies face detection to every frame within a certain interval of frames. This interval should be small enough not to miss important face sequences, yet large enough to ensure reasonable processing time. Optimally, we apply the face detector at intervals of 10 frames. The system uses the neural network-based face detector [10], which detects mostly frontal faces at various sizes and locations. The detected face is output as a rectangular region that includes most of the skin, but excludes the hair and the background. The face detector can also detect eyes; we use only faces in which eyes are successfully detected to ensure that the faces are frontal and close-up. A detected face is tracked bi-directionally in time to obtain a face sequence.

4.1.2 Skin Color Model Extraction/Tracking

Once a face is detected, the system extracts the skin color model. In several cases, researchers used the Gaussian model in (r, g) space ($r = R/(R + G + B), g = G/(R+G+B)$) as a general skin color model for face tracking [11, 12]. Instead, for our research, the Gaussian model in (R, G, B) space is used because this model is more sensitive to brightness of skin color, and thus is much more suitable for the model tailored for each face.

Let F be the detected face region, and $I(x, y)$ be color intensities $[R\ G\ B]^T$ at (x, y). A skin color model consists of a covariance matrix C, a mean M, and a distance d.

$$M = \frac{1}{N} \sum_{(x,y)\in F} I(x, y) \tag{1}$$

$$C = \frac{1}{N} \sum (I(x, y) - M)(I(x, y) - M)^T \tag{2}$$

where N is the number of pixels in F. We used a constant for d. A model is extracted for each detected face, and is used to extract skin candidate pixels in the subsequent frames. A pixel $I(x, y)$ is a skin candidate pixel if $(I(x, y)-M)^T C^{-1}(I(x, y)-M) < d^2$. Then a binary image of the skin candidate pixels is composed, and noise reduction with region enlarging/shrinking and contour tracing of regions is applied to obtain skin candidate regions. The overlap between each of these regions and each of the face regions of the previous frame is evaluated to decide whether one of the skin candidate regions is the succeeding face region. In addition, the scene change detection method based on the sub-region color histogram matching [13] is applied. Face region tracking is continued until a scene change is encountered or until no succeeding face region is found.

4.2 Face Identification

To evaluate face identification, we employed a face similarity measurement based on the eigenface method. Since the eigenface method is very sensitive to face direction, it is desirable to use frontal faces for evaluation. However, detected faces by the method described above are not necessarily frontal enough. On the other hand, as we have face *sequences*, we can choose any face from the sequence. Therefore, we first select the best frontal view of a face, i.e., the *most frontal face* from each face sequence, then apply the eigenface method to the selected faces for identification of face sequences.

4.2.1 The Most Frontal Face Selection

To choose the most frontal face from all detected faces, a face skin region clustering method is first applied. For each detected face, cheek regions, which are sure to have the skin color, are located by using the eye locations. Using the cheek regions as initial samples, region growing in the (R, G, B, x, y) space is applied to obtain the face skin region. We assume a Gaussian model in (R, G, B, x, y) space; (R, G, B) contributes by making the region have skin color, and (x, y) contributes by keeping the region almost circular. Then, the center of gravity (x_f, y_f) of the face skin region is calculated. Let the locations of the right and left eyes of the face be (x_r, y_r), (x_l, y_l), respectively. We assume that the most frontal face has the smallest difference between x_f and $(x_l + x_r)/2$, and the smallest difference between y_l and y_r. To evaluate these conditions, we calculate the frontal factor Fr for every detected face;

$$w_f = \frac{5}{3}(x_l - x_r) \tag{3}$$

$$Fr = 1 - \frac{|2x_f - x_r - x_l|}{w_f} + \frac{1}{2}(1 - \frac{|y_l - y_r|}{w_f}) \tag{4}$$

where w_f is the normalized face region size. The factor for an ideal frontal face is 1.5. The system chooses the face having the largest Fr to be the most frontal face of the face sequence. Figure 5 shows example faces, extracted face skin regions, and frontal factors.

4.2.2 Eigenface-Based Face Identification

We employ the eigenface-based method [5] to evaluate face identification. Each of the most frontal faces is normalized into a 64×64 image using the eye positions, then converted into a point in the 16-dimensional eigenface space. Face identification can be evaluated as the face distance, i.e., the Euclidean distance between two corresponding points in the eigenface space. Let $d_f(F_i, F_j)$ be the face distance of faces F_i and F_j. The similarity $S_f(F_i, F_j)$ is defined as follows:

$$S_f(F_i, F_j) \quad = \quad e^{-\frac{d_f^2(F_i, F_j)}{2\sigma_f^2}} \tag{5}$$

where σ_f is a standard deviation of the Gaussian filter in the eigenface space. The range of similarity is from 0 to 1, where similarity of the same face is 1.

4.3 Evaluation

Figure 6 shows several results of the face extraction method. The start and end frames of a face sequence and the selected frontal face frame are shown. In Figure 6(a), although the faces appearing at the start and end frames are not frontal, the system successfully selected the frontal face. Figure 6(b) shows that the system can handle face sequences having scene changes using special effects (wiping, etc.) in the start and end frames of the sequence. A 30-minute video is processed in roughly 30 hours on an SGI workstation (MIPS R4400 200MHz).

To evaluate face sequence detection, we examined the face sequence extraction results of a half hour news video. The system extracted 65 face sequences, and missed four sequences due to face detection failure (two cases had specular reflection on glasses, and two cases had shade on faces). The system output one non-face sequence due to face detection error, and two sequences each of which is composed of two face sequences merged into one sequence. In one case, the system failed to detect the scene change because it dissolved between two sequences. In another case, the system failed to track the face because it was a monochrome segment. This video was one of the most difficult for face sequence extraction, but the system extracted more than 90% of actual face sequences with one false extraction.

To examine face identification results, we manually named each face sequence. Among 556 face sequences from 5 hours of news videos, we manually named 308 sequences and left 248 unknown. Then we examined every pair of face sequences to obtain distances $d_f(F_i, F_j)$. The distribution of distances of pairs having the same name (identical pairs) and of others (non-identical pairs) was inspected. Figure 7 shows their distribution density graphs. Though these two graphs are not separated completely, the graph of identical pairs has its peak at about 1000, whereas the graph of non-identical pairs has its peak at about 5000. Table 1 shows the statistics of pairs having distances less than 1000, 1500, and 2000, respectively. Column (a) represents the occupancy ratio having distances less than each value among every identical pair, column (b) represents the occupancy ratio having distances less than each value among every non-identical pair, and column (c) represents the occupancy ratio of identical pairs among every pairs having distances less than each value. If we focus on pairs having distances less than 1000, 99% of them are identical and 14% of identical pairs are covered. However, as for pairs having distances less than 2000, even 52% of identical pairs are covered, only 57% of them are identical, i.e., 43% are non-identical. Based on this result, the distance 1000 to 1500 can be a criteria to distinguish identical and non-identical pairs of faces. To prevent non-identical pairs being regarded as identical, we used 1000 for σ_f in Eq. (5) in the experimental Name-It system.

5 Name Information Extraction

This section describes the extraction of names which might correspond to persons of interest in topics. Advanced natural language processing is employed to extract name candidates from transcripts. We will describe how the name candidate extraction uses lexical/grammatical analysis and the knowledge of the structure of a topic in news videos. Finally, the system outputs a three-tuple list: a candidate word, timing information, and a score representing the likelihood of being a name of a person of interest.

5.1 Typical Structure of News Videos

The highest component in news video is an individual topic. Each topic contains one or more paragraphs, which roughly correspond to scenes. In closed-caption texts of CNN Headline News, the components can easily be distinguished; a topic is led by >>>, and a paragraph is led by >> (See Figure 8.). To discriminate an anchor/live video shot from videos, we use this literal information, instead of news video structuring techniques [6, 7]. A typical paragraph at the beginning of the topic is an anchor paragraph, in which an anchor person gives an overview of the topic. Live video paragraphs, which are actual videos related to the topic, or speeches by a person of interest, are typically presented after an anchor paragraph. A live video paragraph, especially one that includes someone's speech, is quite important for Name-It; this paragraph almost certainly contains a close-up scene of that person. However, we should note that the person rarely mentions his/her own name in the speech; thus corresponding transcripts may not contain desired name. The extra care needed to handle this situation is described in the following sub-sections. Finally, each name candidate is output with the score which represents a likelihood factor that the name is likely to correspond to a person of interest.

5.2 Conditions of Name Candidates

Each name candidate should satisfy some of the following conditions:

1. The candidate should be a noun that represents a person's name or that describes a person (president, fireman, etc.).

2. The candidate should preferably be an agent of an act, especially an act of speech, of attendance at a meeting, or of a visit. For example, a speaker is usually centered in the speech scene, while the other people are not always shown in videos even if they are mentioned.

3. The candidate tends to be mentioned earlier than others in the topic in transcripts. (In a news video, important information which might have corresponding images is usually mentioned earlier, rather than later.)

4. The candidate tends to be mentioned just before a live video is shown. The person appearing in a live video rarely mentions his/her own name. Instead, just before the live video, an anchor person tends to introduce him/her (See Figure 8.).

The system evaluates these conditions for each word in the transcripts by using a dictionary (the Oxford Advanced Learner's Dictionary [14]), thesaurus (WordNet [15]), and parser (Link Parser [16]). Then, the system outputs the three-tuple list: a word, timing information (frame), and a normalized score reflecting the above conditions.

5.3 Score Calculation

Referring to the dictionaries and the parsing results, the system calculates the score for each word in transcripts. The score is normalized so that a score close to 1.0 corresponds to a word which most likely corresponds to a person of interest. The score calculation is defined as follows:

Grammatical Score: After consulting the dictionary, the system gives 1.0 to proper nouns, 0.8 to common nouns, and 0 to other words. And by consulting the parsing results, the system gives 1.0 to nouns, and 0.5 to other words; if the system fails to parse, it gives 0.5 to all words. The net grammatical score is the product of the two.

Lexical Score: After consulting the thesaurus, the system gives 1.0 to persons, 0.8 to social groups, and 0.3 to other words.

Situational Score: The act corresponding to the word is represented by the verb in the sentence which includes the word. By looking the verb up in the thesaurus, the system gives 1.0 to speech, 0.8 to attendance at meetings, and 0.3 otherwise.

Positional Score: The system gives 1.0 to words that appear in the first sentence of a topic, 0.5 to words that appear in the last sentence of a paragraph, and linearly interpolated score to other words according to the position of the sentence where the word appears. As for live video paragraphs, the system also outputs the same tuples as those of the paragraph which appears before the live video paragraphs (possibly the anchor paragraph), replacing the timing information with that of the live video (See Figure 8.). In addition, it replaces the positional score according to the position of the sentence in the anchor paragraph: 1.0 for the sentence just before the live video, 0.5 for the first sentence of the topic, and a linearly interpolated score otherwise.

Finally, the net score is calculated as the product of all 4 scores. The execution time for a 30-minute news video is approximately 1.5 hour on an SGI workstation (MIPS R4400 200MHz). Most of that time is consumed by parsing.

5.4 Evaluation

We examined one 30-minute news video and manually extracted 105 name words from a transcript containing 3462 words. While the system automatically extracted 752 words as name candidates, 94 of them were correct (9 were missed, and 658 were false alarms.). This excessive name candidate extraction is not satisfactory. But this is because the system extracts words which are proper nouns OR nouns used as agents, in order not to miss any “name.”

6 Video Caption Recognition

Figure 9 shows a typical frame with video captions. Since we use CNN Headline News for target news videos, captions are shown in bright color, directly onto the background images, and using Roman or Gothic font types. Captions in Roman font mainly represent names of persons or places, and captions in Gothic font mainly represent titles of persons. Thus we use Roman font captions for Name-It. To achieve video caption recognition, we need text region detection, and character recognition, including image enhancement, font type identification, and character region extraction. Since character recognition results cannot be perfect, we also employ an inexact match for recognition results with words.

6.1 Text Area Detection

A typical text region can be characterized as a horizontal rectangular structure of clustered sharp edges, because characters usually form regions of high contrast against the background. By detecting these properties, we extract regions from video frames that contain textual information. Figure 10 illustrates the process of detecting text; primarily, regions of horizontal titles and captions.

We first apply a 3×3 horizontal differential filter to the entire image with appropriate binary thresholding for extraction of vertical edge features. Smoothing filters are then used to eliminate extraneous fragments, and to connect character sections that may have been detached. Individual regions are identified by cluster detection and their bounding rectangles are computed. Clusters with bounding regions that satisfy the following constraints are selected:

- Cluster Size $>$ 70 pixels,

- Cluster Fill Factor $\geq$ 0.45,
- Horizontal-Vertical Aspect Ratio $\geq$ 0.75.

A cluster's bounding region must have a large horizontal-to-vertical aspect ratio as well as satisfying various limits in height and width. The fill factor of the region should be high to insure dense clusters. The cluster size should also be relatively large to avoid small fragments. An intensity histogram of each region is used to test for high contrast. This is because certain textures and shapes appear similar to text but exhibit low contrast when examined in a bounded region. Finally, consistent detection of the same region over a certain period of time is also tested since text regions are placed at the exact position for many video frames. Figure 11 shows detection examples of words.

6.2 Character Recognition

There are two essential difficulties in video caption recognition: complicated backgrounds and insufficient image resolution. We preprocess the detected text regions to overcome these difficulties and adapt a conventional character recognition method to video captions.

Since captions are superimposed on news video, character extraction suffers from complicated backgrounds with movement. On the other hand, video captions are placed at the same position for sequence frames. Therefore, minimizing intensities among frames works as an enhancement of characters in video captions, assuming that characters have high intensities (See Figure 12.). Enhanced image $I_e(x,y)$ is obtained from input frames, $I_i(x,y), I_{i+1}(x,y), \cdots, I_{i+n}(x,y)$ as follows:

$$I_e(x,y) \quad = \quad min(I_i(x,y), I_{i+1}(x,y), \ldots, I_{i+n}(x,y)) \tag{6}$$

where i and $i+n$ are respectively the start frame number and the end frame number of the text block, and (x,y) indicates the position of a pixel.

In CNN Headline News, characters of the video caption are small to avoid occlusion of interesting objects. Therefore, image resolution of characters is insufficient for recognition. To obtain clearer images, we apply a linear interpolation technique to quadruple the resolution of the image (original: $N \times N \longrightarrow$ result: $4N \times 4N$). Each pixel of the original image $I_e(x,y)$ is placed at every fourth pixel in both the x- and y-direction in the high resolution image $I_h(x,y)$: $I_h(4x,4y) =$

$I_e(x, y)$. Other pixels are obtained by linear interpolation using neighboring pixel values as follows:

$$I_h(x,y) = \frac{\sum\limits_{(x_0,y_0)\in\mathcal{N}(x,y)} w(x-x_0, y-y_0)\cdot I_e(\frac{x_0}{4},\frac{y_0}{4})}{\sum\limits_{(x_0,y_0)\in\mathcal{N}(x,y)} w(x-x_0, y-y_0)} \qquad (7)$$

where $\mathcal{N}(x,y) = \{(x_0,y_0)|x_0 \in \{\lfloor\frac{x}{4}\rfloor\cdot 4, \lceil\frac{x}{4}\rceil\cdot 4\}, y_0 \in \{\lfloor\frac{y}{4}\rfloor\cdot 4, \lceil\frac{y}{4}\rceil\cdot 4\}\}$ and $w(x,y) = ||(x,y)||^{-1}$. Figure 13 shows an example of high resolution image creation.

To create a binary image of the characters, thresholding at a fixed value is applied. Then, we use horizontal and vertical projections for the binary image to extract the bounding rectangular block of each character. In our video data, the threshold level should be changed according to font types. In our example, the threshold level for Roman font text blocks should be lower than the threshold level for Gothic font text blocks. We use the filling factor of text blocks to distinguish font types. After font type estimation, character extraction is repeated using the revised threshold value.

If the width of a bounding rectangle exceeds the upper limit of the widest character, the block is decomposed into small regions according to the width of character reference patterns. This decomposition and subsequent reference pattern matching is repeated until the best combination of characters is achieved.

Extracted characters are normalized to a fixed size and blurred into gray scale images. This processing makes the character matching robust to change in thickness and position. The normalized gray scale data $I_n(x, y)$ is matched with the reference pattern $I_r(x, y)$ of each letter based on a correlation metric. The matching metric m_c is described as follows:

$$m_c = \frac{\sum\limits_{(x,y)} I_n(x,y)\cdot I_r(x,y)}{\sqrt{\sum\limits_{(x,y)}(I_n(x,y))^2}\sqrt{\sum\limits_{(x,y)}(I_r(x,y))^2}}. \qquad (8)$$

The reference pattern $I_r(x, y)$, having the largest metric m_c, is selected as the first candidate. In the same manner, second and third candidates are selected. Recognition rates evaluated with a 30-minute video (95 words, 635 characters) are shown in Table 2.

6.3 Modified Edit Distance

To match video caption recognition results with names, we define a similarity, which is based on the edit distance [17]. Assume that x is the character recognition result of a word. Let $x(i)$ be the i-th letter of the recognition result x, and $x(i)^c_j$ and $x(i)^s_j$ be the resultant character and its score of the letter of the j-th precedence, respectively. $x(i)^c_1$ is the first preceded character of the letter $x(i)$. Likewise, let y be a word to be matched with, and $y(i)$ be the i-th letter of y. In addition, $x(i..)$ denotes the substring of x begins at i-th position. The modified edit distance $d_c(x, y)$ is defined recursively as follows:

$$\begin{aligned} d_c(x,y) &= min \left\{ \begin{array}{l} 1 + d_c(x(2..), y), \\ 1 + d_c(x, y(2..)), \\ c_c(x(1), y(1)) + d_c(x(2..), y(2..)) \end{array} \right\} \quad (9) \\ d_c(\varepsilon, \varepsilon) &= 0 \end{aligned}$$

where ε represents a null string. $c_c(p, q)$ is the cost function between characters p and q, which is defined as follows:

$$c_c(p,q) = \left\{ \begin{array}{ll} 1 & (\forall_i p^c_i \neq q) \\ 1 - p^s_i / p^s_1 & (p^c_i = q) \end{array} \right. \quad (10)$$

The distance is calculated using a dynamic programming algorithm.

Then, the normalized distance $\hat{d_c}(x, y)$ between x and y is defined as follows:

$$\hat{d_c}(x,y) = \frac{d_c(x,y)}{max(len(x), len(y))} \quad (11)$$

where $len(x)$ returns the length of the string x. When x and y are the same, the distance is 0, whereas when x and y are totally different (i.e., x and y do not share any character), the distance is 1.

7 Face-Name Association

7.1 Algorithm

In this section, the algorithm for retrieving face candidates by a given name is described. We use the co-occurrence factor [1, 2] taking advantage of face extraction/identification, name extraction, and video caption recognition. Let N and

F be a name and a face, respectively. The co-occurrence factor $C^*(N, F)$ measures the degree of how well face F will match name N. Think of the names $N_a, N_b, \cdots$ and the faces $F_p, F_q, \cdots$, and N_a corresponds to F_p. Then $C^*(N_a, F_p)$ should have the largest value among co-occurrence factors of any combinations of N_a and the other faces (e.g., $C^*(N_a, F_q)$, etc.), or of the other names and F_p (e.g., $C^*(N_b, F_p)$, etc.). Retrieval of face candidates by a given name is realized using the co-occurrence factor:

1. Calculate co-occurrences of combinations of all face candidates with a given name.
2. Sort co-occurrences.
3. Output faces that correspond to the N largest co-occurrences.

Retrieval of name candidates by a face is also realized.

7.2 Co-occurrence Factor

Extracted face sequences are obtained as a two-tuple list (timing, face identification): $\{(t_{F_i}, F_i)\} = \{(t_{F_1}, F_1), (t_{F_2}, F_2), ...\}$, where $t_{F_i} = t_{start,F_i} \sim t_{end,F_i}$. We can define the duration of a face sequence by the function $dur(t_{F_i}) = t_{end,F_i} - t_{start,F_i}$. The occurrence of a face F in a video can be expressed as the occurrence function $o_f(t; t_F)$:

$$o_f(t; t_F) = \begin{cases} e^{-\frac{|t - t_{start,F}|^2}{2\sigma_t{}^2}} & (t < t_{start,F}) \\ 1 & (t_{start,F} \leq t < t_{end,F}) \\ e^{-\frac{|t_{end,F} - t|^2}{2\sigma_t{}^2}} & (t_{end,F} \leq t) \end{cases} \quad (12)$$

This is basically a step function having 1 in the range between $t_{start,F}$ and $t_{end,F}$, but its edges are dispersed using a Gaussian filter with standard deviation σ_t. This function is intended to represent the likelihood of occurrence of the name of F in the transcript. The Gaussian filter is expected to compensate for time delay between the video and transcript. We define the extended face occurrence function $O_f(t; F)$, taking into account face similarities:

$$O_f(t; F) = \sum_i S_f(F_i, F) o_f(t; t_{F_i}). \quad (13)$$

Name extraction results are given as a three-tuple list (word, timing, score): $\{(N_j, t_{N_j,k}, s_{N_j,k})\} = \{(N_1, t_{N_1,1}, s_{N_1,1}), (N_1, t_{N_1,2}, s_{N_1,2}), \ldots (N_2, t_{N_2,1}, s_{N_2,1}), \ldots\}$. Note that a name N_j may occur several times in a video, so each occurrence is indexed by k. Since name occurrence does not have duration, each name occurrence can be expressed as a Dirac delta function $\delta(t)$. Then the name occurrence function $O_n(t; N)$, which represents occurrence of a name N is defined as follows:

$$O_n(t; N) = \sum_k s_{N,k} \delta(t - t_{N,k}). \tag{14}$$

By multiplying the delta function by score $s_{N,k}$, the function represents occurrences of the name which may correspond to a person of interest. A typical configuration of face and name occurrence functions is shown in Figure 14.

Next, timing similarity, $S_t(t_F, t_N)$ of a face F and a name N is defined as: $S_t(t_F, t_N) = o_f(t_N; t_F)$. The co-occurrence factor $C(N, F)$ of the name N and the face F is defined as follows:

$$C(N, F) = \frac{\int_0^\infty O_f(t; F) \cdot O_n(t; N) dt}{\sqrt{\int_0^\infty (O_f(t; F))^2 dt \int_0^\infty sq(O_n(t; N)) dt}} \tag{15}$$

$$\simeq \frac{\sum_i S_f(F_i, F) \sum_k s_{N,k} S_t(t_{F_i}, t_{N,k})}{\sqrt{\sum_i S_f^2(F_i, F) dur(t_{F_i}) \sum_k s_{N,k}^2}} \tag{16}$$

where $sq(f) = f \cdot (f \otimes \Delta)$, $\Delta(t) = 1$ if $|t| < \varepsilon$, 0 otherwise. $\otimes$ represents convolution, and ε is sufficiently small. Intuitively, the numerator represents the number of occurrences of the name N that coincide with face F, taking face similarities and name scores into account. It is normalized with the denominator to prevent the "anchor person problem" (An anchor person coincides with almost any name. A face/name that coincides with any name/face should correspond to NO name/face.).

7.3 Incorporation of Video Caption Recognition Results

We extend the co-occurrence factor definition by utilizing video caption recognition results. The caption recognition results are obtained as a two-tuple list (timing, recognition result): $\{(t_{C_i}, C_i)\} = \{(t_{C_1}, C_1), (t_{C_2}, C_2), ...\}$, where $t_{C_i} = t_{start,C_i} \sim t_{end,C_i}$ because each caption has duration. First, the caption recognition result is chronologically compared with a face. We simply define the timing similarity, $S'_t(t_C, t_F)$ of a caption C and a face F, as follows:

$$S'_t(t_C, t_F) = \begin{cases} 0 & (t_{end,F} < t_{start,C} \text{ or } t_{end,C} < t_{start,F}) \\ 1 & (\text{otherwise}) \end{cases} \tag{17}$$

Next, the similarity between a caption recognition result C and a name N is evaluated using the distance $\hat{d}_c(C, N)$. We use only pairs of captions and names that are quite similar. Thus the similarity $S_c(C, N)$ of a caption C and a name N is defined as follows:

$$S_c(C, N) = \begin{cases} 0 & (\hat{d}_c(C, N) > \theta_c) \\ 1 & (\text{otherwise}) \end{cases} \tag{18}$$

where θ_c is the threshold value for the distance between captions and names. θ_c is set to 0.2 for our experimental system.

Then we define the extended co-occurrence factor $C^*(N, F)$ of a name N and a face F taking advantage of video caption recognition results as follows:

$$C^*(N, F) = \frac{\sum_i S_f(F_i, F)(\sum_k s_{N,k} S_t(t_{F_i}, t_{N,k}) + *)}{\sqrt{\sum_i S_f^2(F_i, F) dur(t_{F_i}) \sum_k s_{N,k}^2}} \tag{19}$$

$$* = w_c \sum_j S'_t(t_{C_j}, t_{F_i}) S_c(C_j, N) \tag{20}$$

where w_c is the weight for caption recognition results. Roughly speaking, when a name and a caption match and the caption and a face match at the same time, the face equivalently coincides with w_c occurrences of that name. We use 1 for the value of w_c.

8 Experiments

The Name-It System was implemented on an SGI workstation. We processed 10 CNN Headline News videos (30 minutes each), i.e., a total of 5 hours of video. The system extracted 556 face sequences from videos. Name-It performs name candidate retrieval from a given face, and face candidate retrieval from a given name. In face-to-name retrieval, the system is given a face, then outputs name candidates with co-occurrence factors in descending order. Likewise, in name-to-face retrieval, the system outputs face candidates of a given name with co-occurrence factors in descending order.

Figure 15 shows the results of face-to-name retrieval. In each result, an image of a given face and ranked name candidates associated with co-occurrence factors are shown. Correct answers are shown with a circled ranking number. Figure 16 shows the results of name-to-face retrieval. The top-4 face candidates are shown in order from left to right with corresponding co-occurrence factors. These results demonstrate that Name-It achieves effective face-to-name and name-to-face retrieval with actual news videos.

Then we evaluate Name-It system in terms of accuracy. We use 308 manually named face sequences (Section 4.3) as the correct answer. Figure 17 and Figure 18 depict accuracy of face-to-name and name-to-face retrieval. In this accuracy evaluation, if (one of) correct answer is output in the top-N candidates, we regard this output of Name-It as correct (the output is correct with N allowed candidates). (Note that a name may correspond to several identical faces, and a face may correspond to given name and family name.) Thus these graphs represent relations between accuracy and the number of allowed candidates. They also show results using both name scores and video caption recognition, results without name scores (set all scores 1.0), results without video caption recognition (set w_c 0), and results without either name scores or video caption recognition.

By comparing results using both name scores and video captions and results without video captions for both graphs, we can say that video caption recognition contributes towards higher accuracy. Actually, there are some faces not being mentioned in the transcripts, but described in video captions. These faces can be named only by incorporating video caption recognition (e.g., Figure 15(d) and Figure 16(e)). Figure 17 depicts that name score evaluation is effective for Face-to-Name retrieval, however, according to Figure 18, it does not cause major difference in accuracy for Name-to-Face retrieval. This result means that name scores properly reflect whether each word corresponds to a person of interest in topics (in Face-to-Name retrieval). By contrast, name scores cannot represent which occurrence of a certain word coincides with a face sequence of the person of the name (in Name-to-Face retrieval). In other words, name scores succeed in inferring which word is likely to correspond to a person of interest. However, they fail to infer which word actually coincides with the face sequence. The main reason of this is the fact that transcripts do not directly explain videos. To overcome this problem, the system may need in-depth transcript recognition, as well as in-depth scene understanding, and a proper way to integrate these analysis results. The graphs also disclose that Name-It achieves accuracy of 33% in face-to-name retrieval, and 46% in name-to-face retrieval with 5 allowed candidates.

9 Conclusions

This paper describes Name-It, a system that associates faces and names in news videos. To accomplish this task, the system integrates face sequence extraction/identification, name extraction, and video caption recognition. Name-It integrates these techniques into a unified factor: co-occurrence. The successful experimental results demonstrate the effectiveness of a multi-modal approach in video content extraction. In addition, the performance of each technology is evaluated. Though the performance of each technology is not always high, Name-It achieves good face-name association as shown in the experiments. Further research will be directed at enhancing each technique, as well as analyzing and improving the integration method.

Acknowledgement

This material is based upon work supported by the National Science Foundation under Cooperative Agreement No. IRI-9411299. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] S. Satoh and T. Kanade, Name-It: Association of face and name in video, in *Proceedings, Computer Vision and Pattern Recognition*, 1997.

[2] S. Satoh, Y. Nakamura, and T. Kanade, Name-It: Naming and detecting faces in video by the integration of image and natural language processing, in *Proceedings, International Joint Conference on Artificial Intelligence*, 1997.

[3] H. D. Wactlar, T. Kanade, M. A. Smith, and S. M. Stevens, Intelligent access to digital video: The informedia project, *IEEE Computer*, **29**, 1996, 46–52.

[4] A. Pentland, R. W. Picard, and S. Schlaroff, Photobook: Content-based manipulation of image databases, *International Journal of Computer Vision*, **18**, 1996, 233–254.

[5] M. Turk and A. Pentland, Eigenfaces for recognition, *Journal of Cognitive Neuroscience*, **3**, 1991, 71–86.

[6] D. Swanberg, C. F. Shu, and R. Jain, Knowledge guided parsing in video database, in *Proceedings, Symposium on Electric Imaging, Science and Technology*, IS&T/SPIE, 1993.

[7] S. W. Smoliar and H. Zhang, Content-based video indexing and retrieval, *IEEE Multimedia*, Summer 1994, 62–72.

[8] R. Chopra and R. K. Srihari, Control structures for incorporating picture — specific context in image interpretation, in *Proceedings, International Joint Conference on Artificial Intelligence*, 1995.

[9] *Proceedings, Sixth Message Understanding Conference*, 1995.

[10] H. Rowley, S. Baluja, and T. Kanade, Human face detection in visual scenes, Tech. Rep. CMU-CS-95-158, School of Computer Science, Carnegie Mellon University, 1995.

[11] J. Yang and A. Waibel, Tracking human faces in real-time, Tech. Rep. CMU-CS-95-210, School of Computer Science, Carnegie Mellon University, 1995.

[12] H. M. Hunke, Locating and tracking of human faces with neural networks, Tech. Rep. CMU-CS-94-155, School of Computer Science, Carnegie Mellon University, 1994.

[13] M. Smith and T. Kanade, Video skimming for quick browsing based on audio and image characterization, Tech. Rep. CMU-CS-95-186, School of Computer Science, Carnegie Mellon University, 1995.

[14] The Oxford Text Archive. http://ota.ox.ac.uk/.

[15] G. Miller, WordNet: An on-line lexical database, *International Journal of Lexicography*, **3**, 1990.

[16] D. Sleator, Parsing english with a link grammar, in *Proceedings, Third International Workshop on Parsing Technologies*, 1993.

[17] P. A. V. Hall and G. R. Dowling, Approximate string matching, *ACM Computing Surveys*, **12**, 1980, 381–402.

Tables

	(a)	(b)	(c)
$<$1000	14%	0.0%	99%
$<$1500	32%	0.2%	77%
$<$2000	52%	0.8%	57%

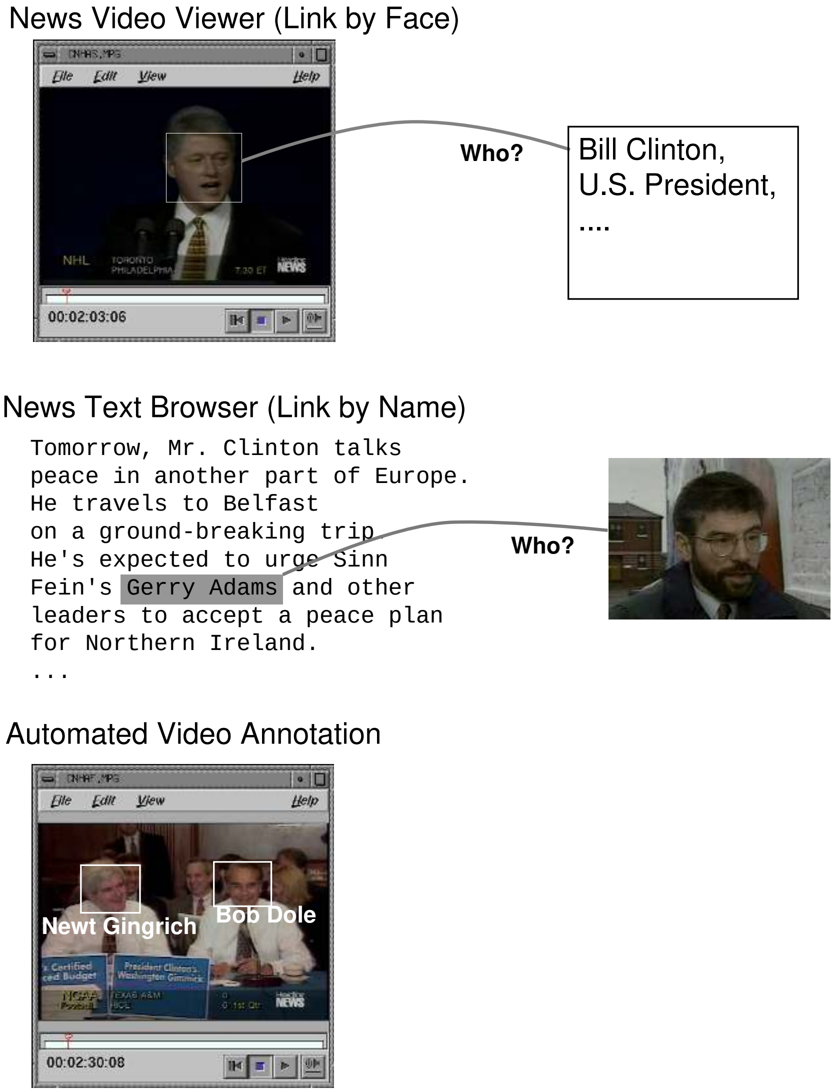
Table 1: Distribution of Face Distance

(a) represents the occupancy ratio having distances less than each value among every identical pair, (b) represents the occupancy ratio having distances less than each value among every non-identical pair, and (c) represents the occupancy ratio of identical pairs among every pairs having distances less than each value.

	Roman	Gothic	R+G
character rate	65%	85%	76%
word rate	22%	74%	48%

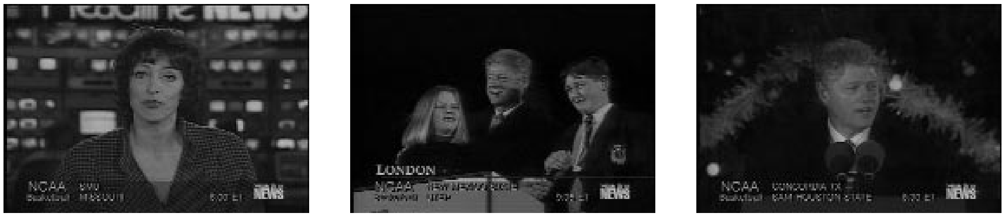
Table 2: Video Caption Recognition Rate

Figures



Tomorrow, Mr. Clinton talks
peace in another part of Europe.
He travels to Belfast
on a ground-breaking trip.
He's expected to urge Sinn
Fein's Gerry Adams and other
leaders to accept a peace plan
for Northern Ireland.
...

Figure 1: Potential Applications of Name-It



MR. CLINTON VISITED NORTHERN IRELAND AND...

Reporter: MR.CLINTON LIGHTED THE CHRISTMAS TREE, ...

I PLEDGE YOU AMERICA’S SUPPORT...
...
Reporter: PRESIDENT CLINTON PLEDGED THE HELP OF U.S. INVESTIMENT...

Figure 2: Typical Composition of News Topic

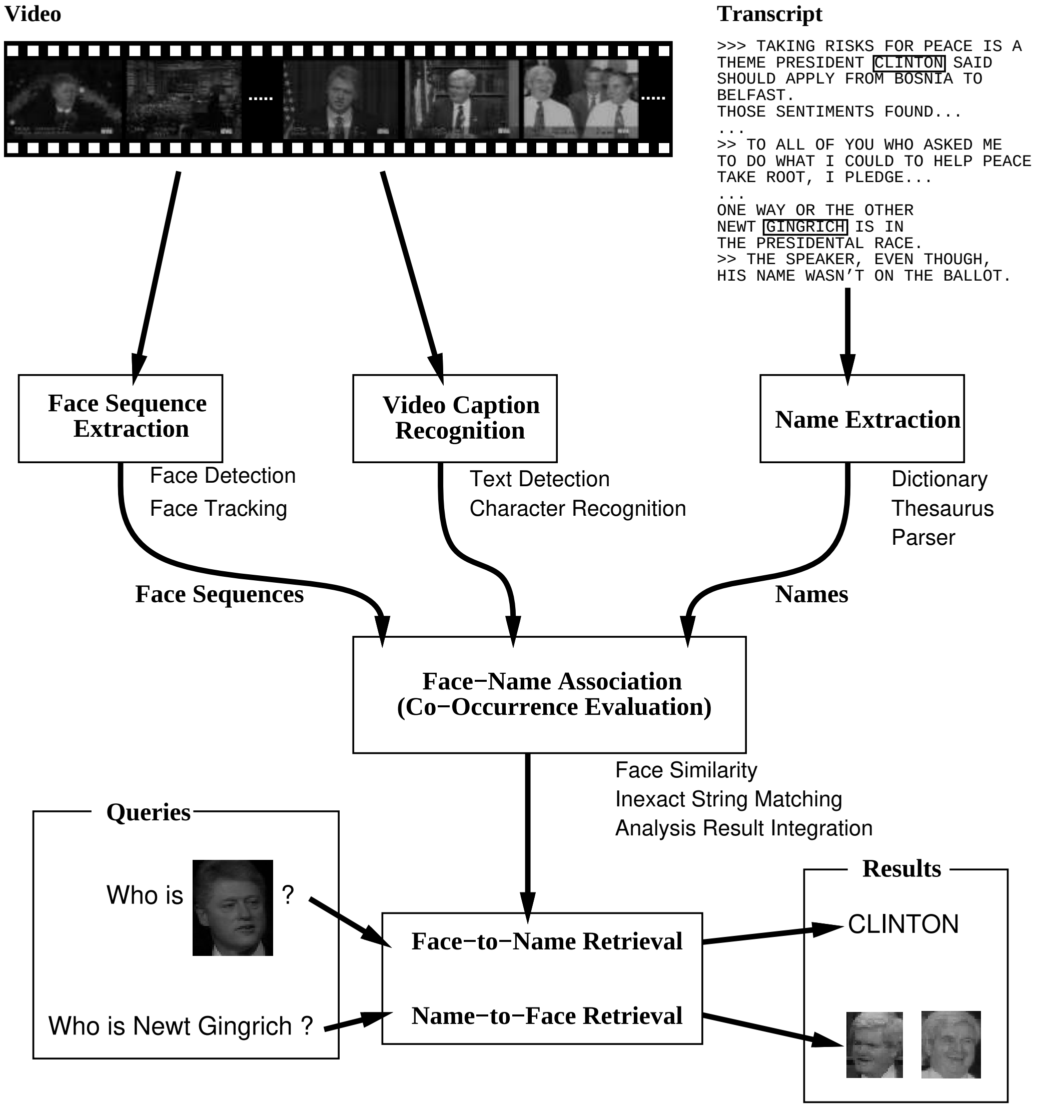


Figure 3: Architecture of Name-It

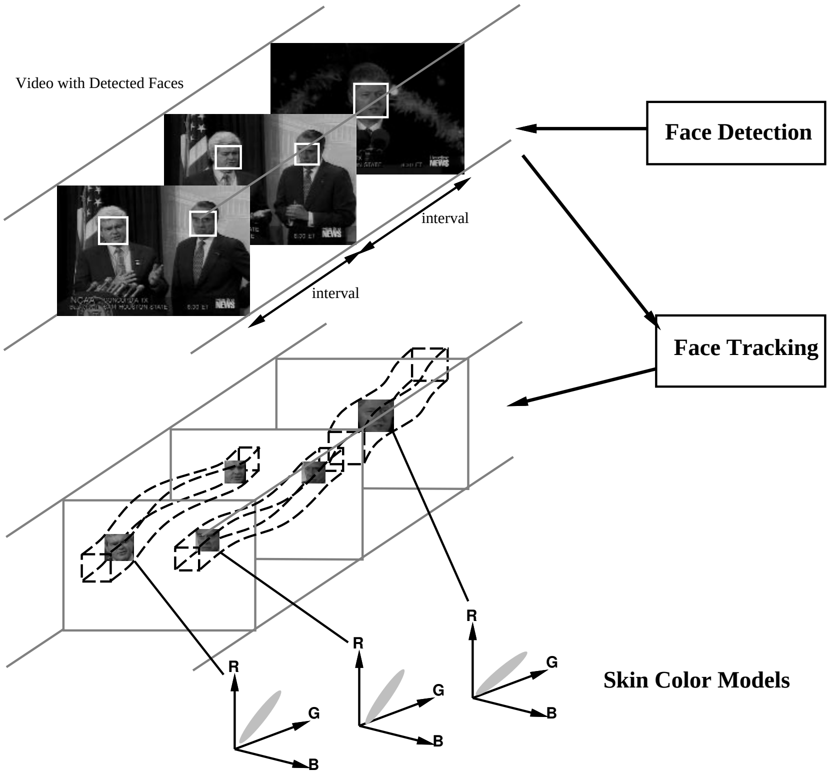


Figure 4: Face Tracking

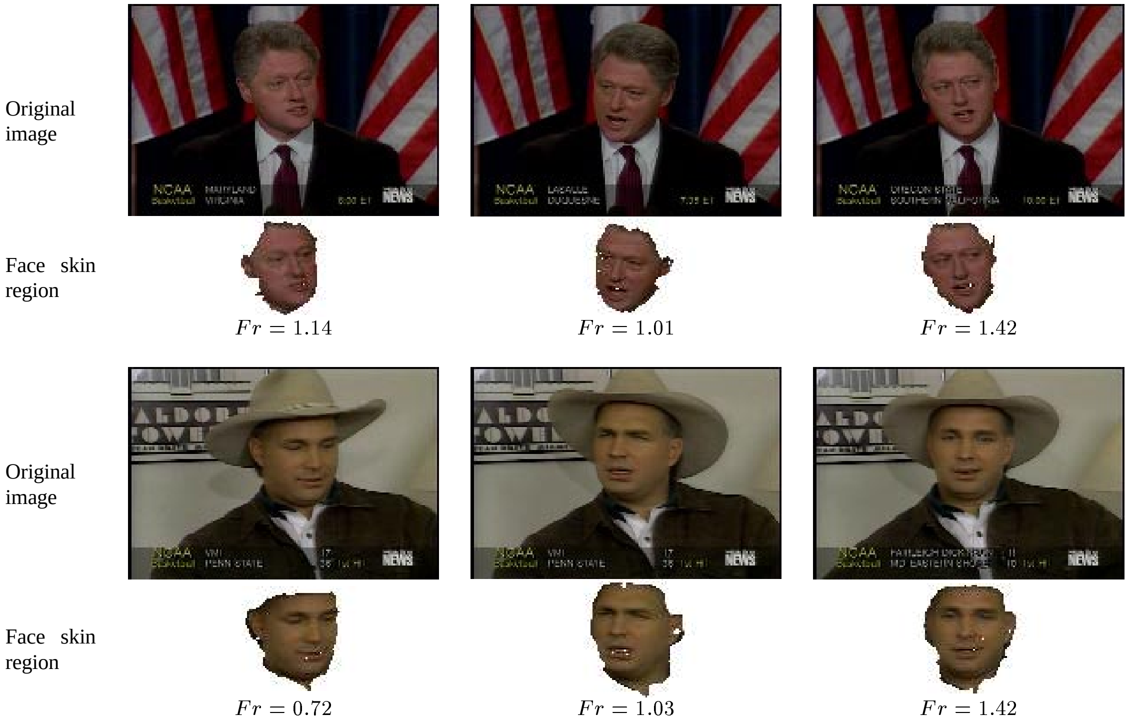


Figure 5: Frontal Face Selection

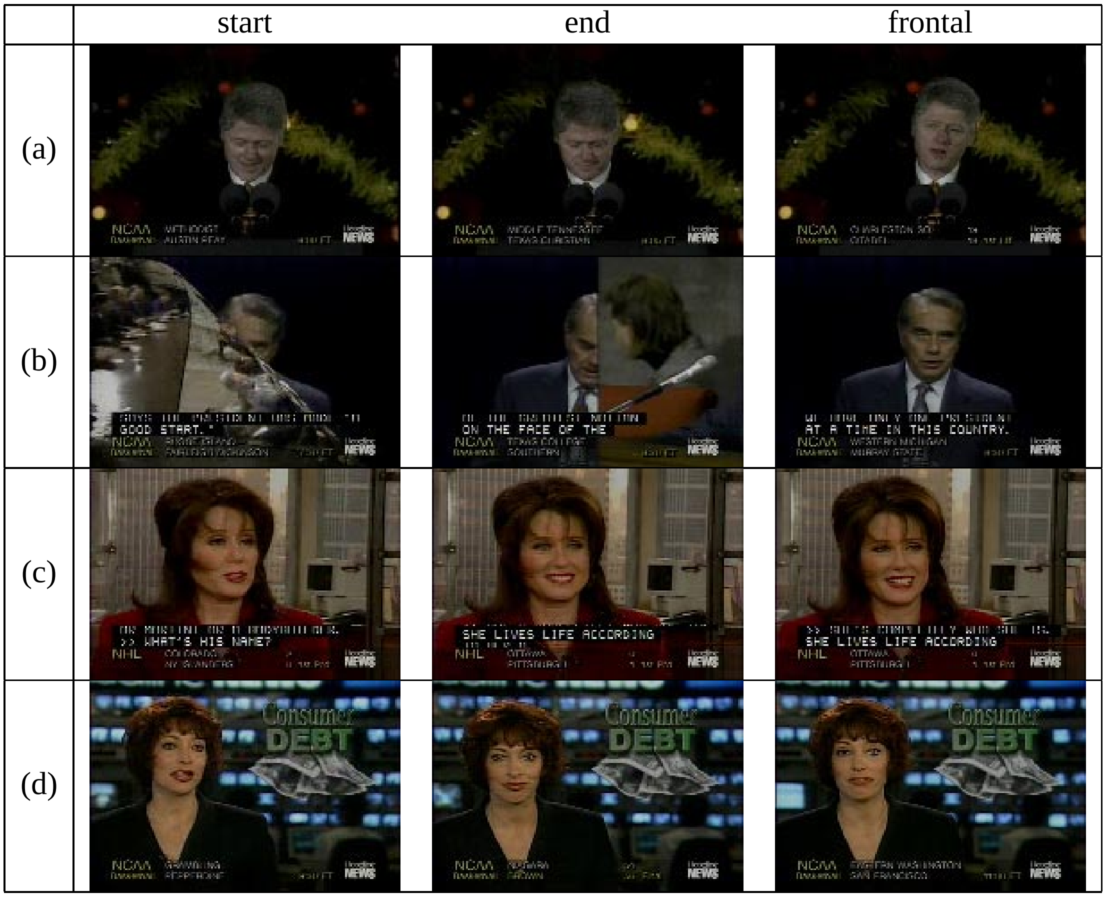


Figure 6: Face Extraction Results

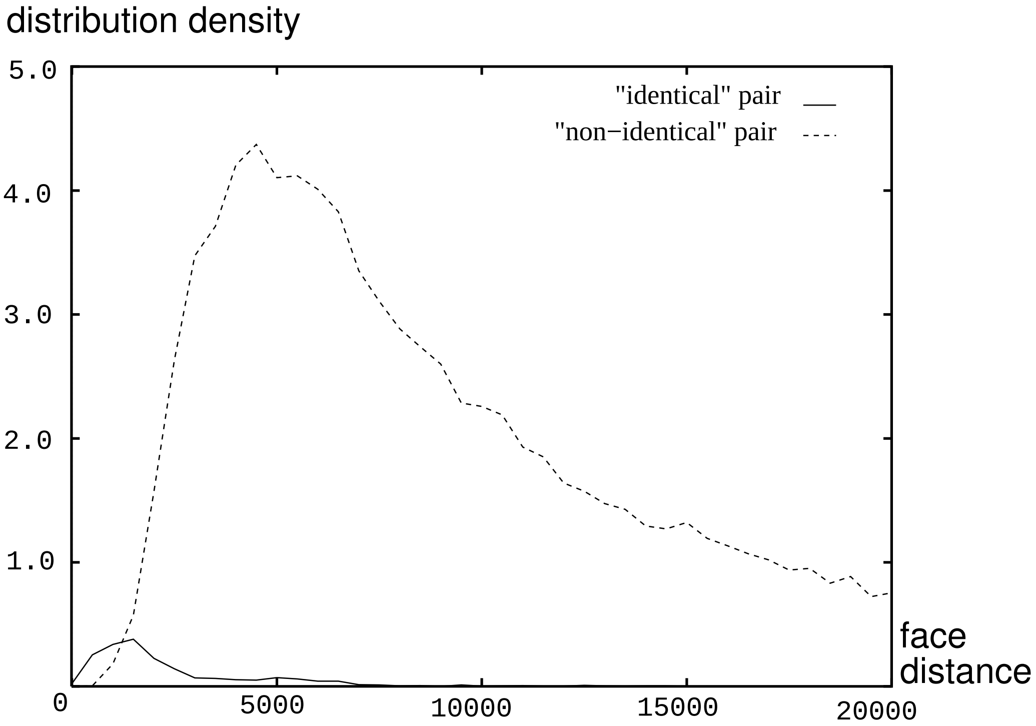


Figure 7: Distribution Density of Face Distance

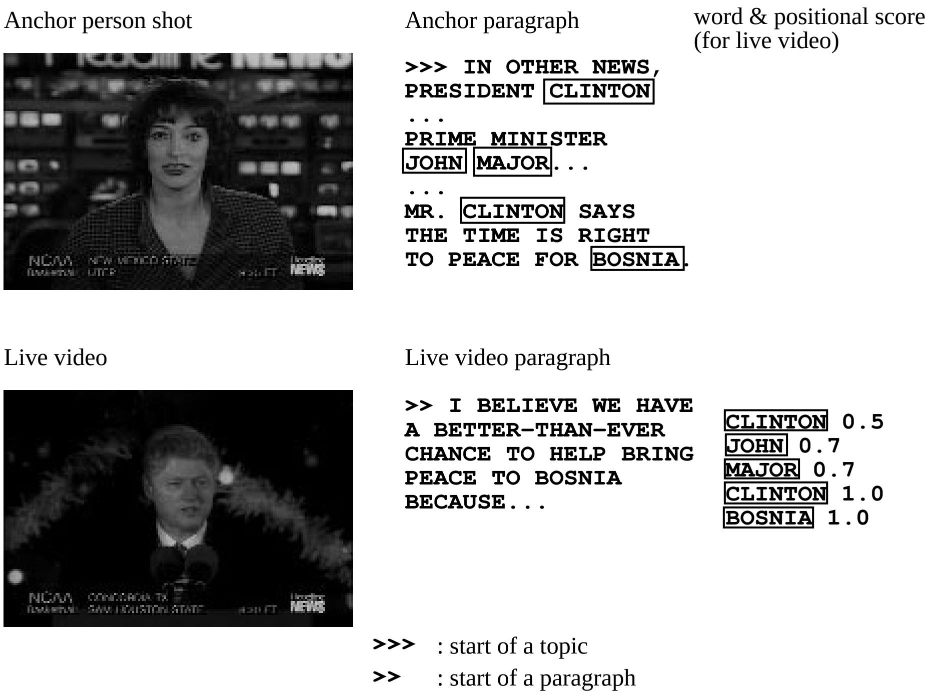


Figure 8: Positional Score for Live Video

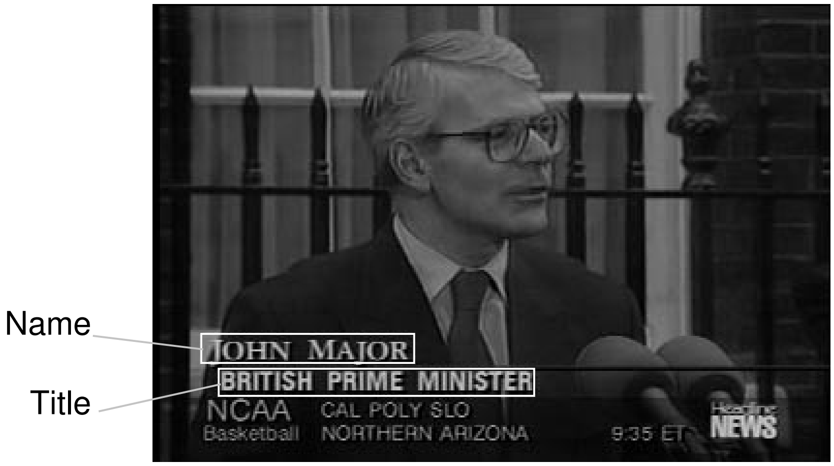


Figure 9: Typical Video Caption

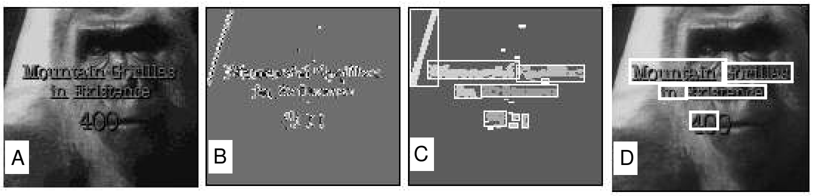


Figure 10: Stages of text detection: A) Input, B) Filtering, C) Clustering, and D) Region Extraction.

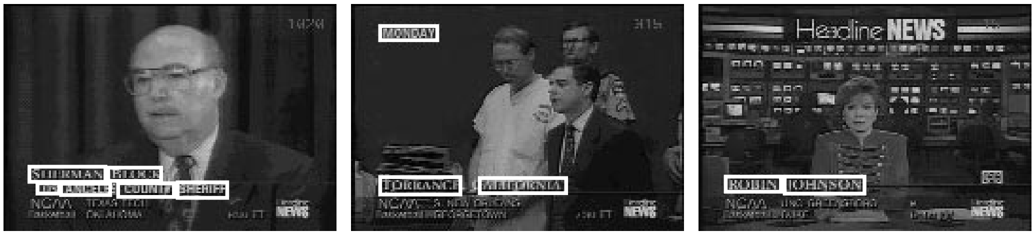


Figure 11: Text detection results with various images.
CNN Sports Ticker portion (bottom portion) was eliminated from processing.

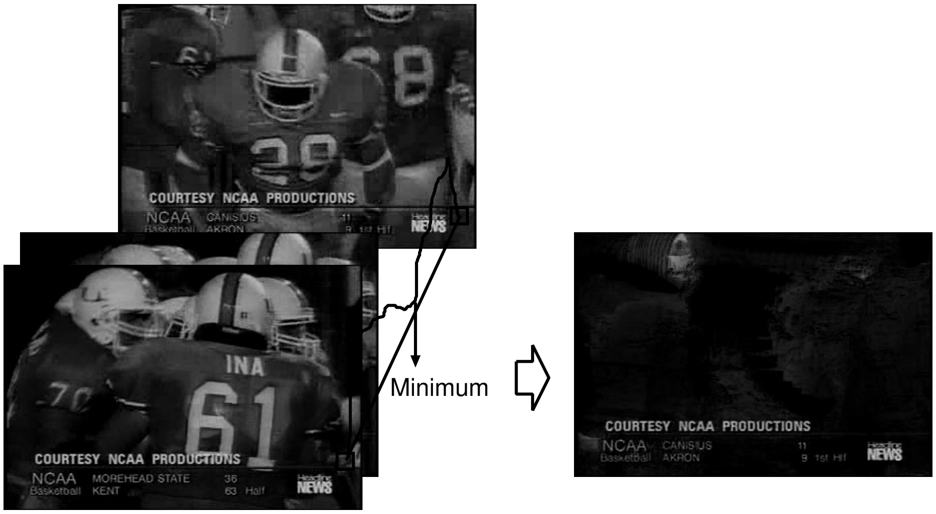


Figure 12: Enhancement of Characters

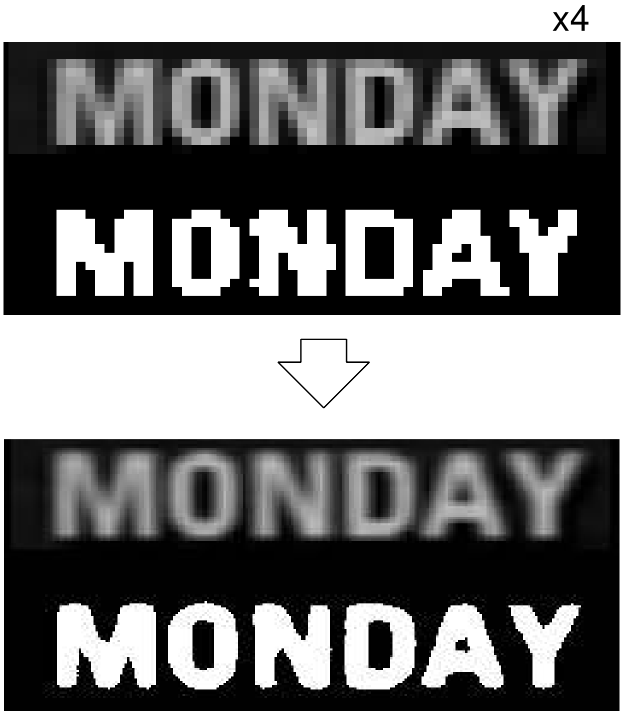


Figure 13: High Resolution Image Creation
Original image/binary image, high resolution image/binary image

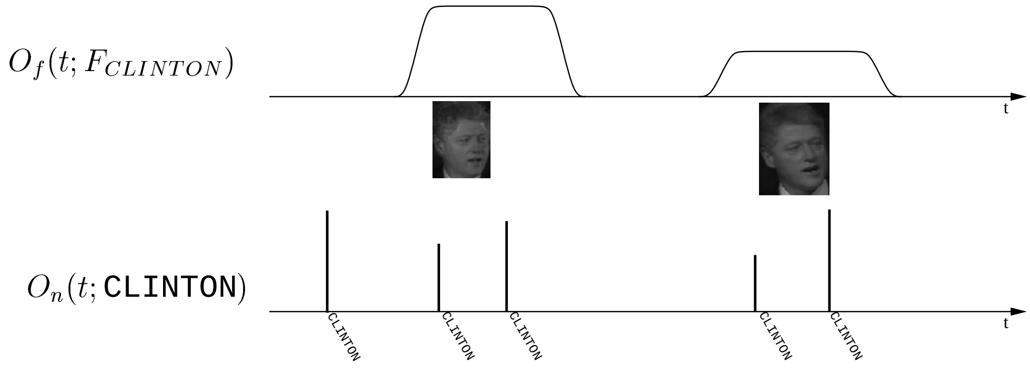
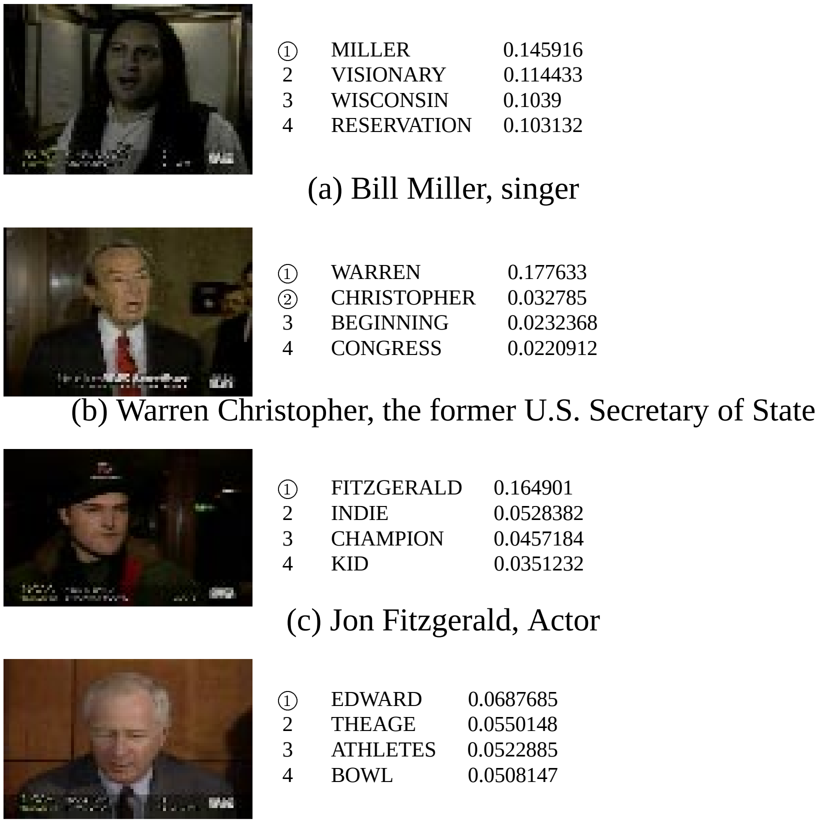


Figure 14: Face and Name Occurrence Function



①	MILLER	0.145916
2	VISIONARY	0.114433
3	WISCONSIN	0.1039
4	RESERVATION	0.103132

(a) Bill Miller, singer

①	WARREN	0.177633
②	CHRISTOPHER	0.032785
3	BEGINNING	0.0232368
4	CONGRESS	0.0220912

(b) Warren Christopher, the former U.S. Secretary of State

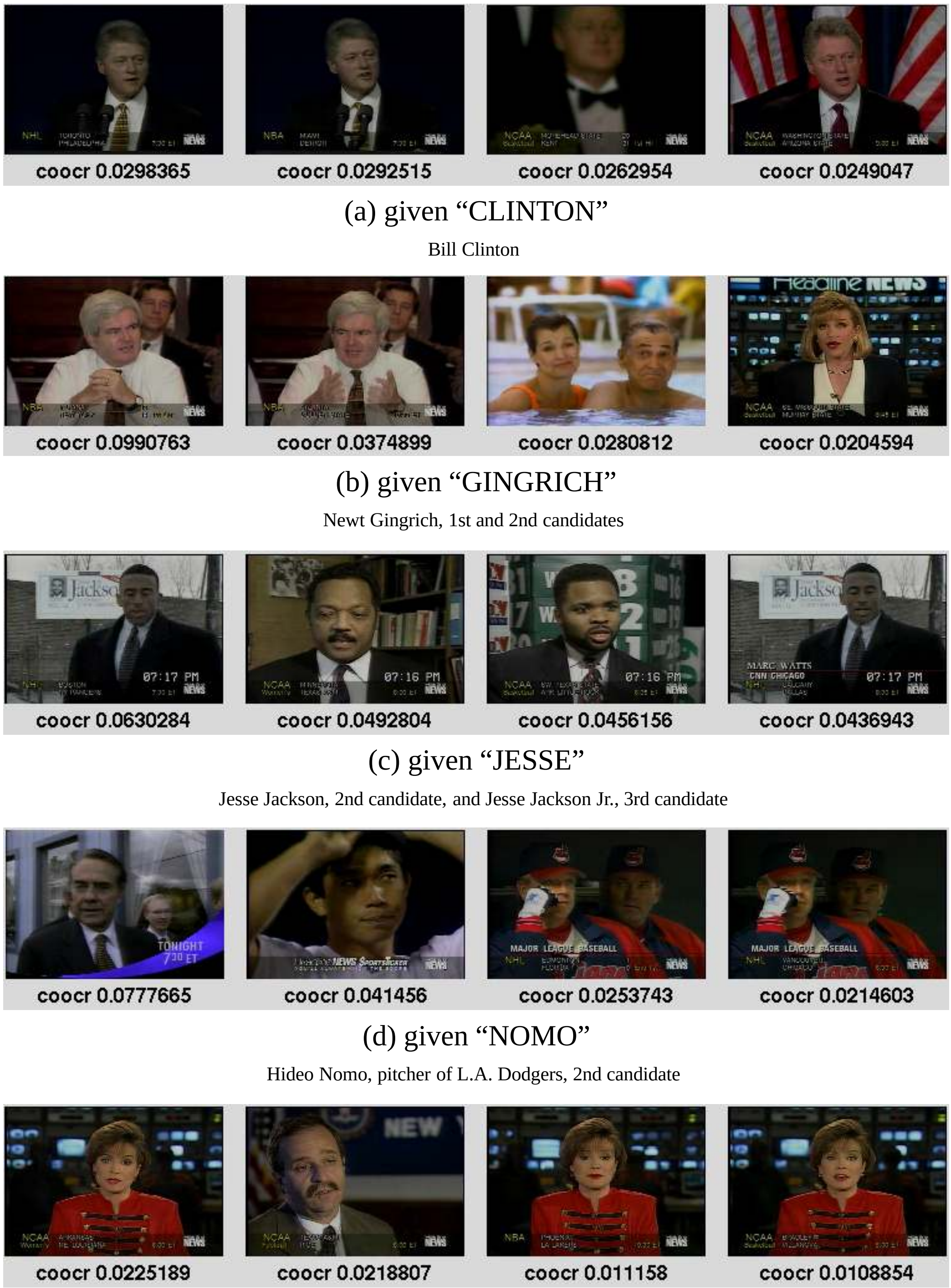
①	FITZGERALD	0.164901
2	INDIE	0.0528382
3	CHAMPION	0.0457184
4	KID	0.0351232

(c) Jon Fitzgerald, Actor

①	EDWARD	0.0687685
2	THEAGE	0.0550148
3	ATHLETES	0.0522885
4	BOWL	0.0508147

(d) Edward Foote, University of Miami President

Figure 15: Face-to-Name Retrieval



(a) given “CLINTON”

Bill Clinton

(b) given “GINGRICH”

Newt Gingrich, 1st and 2nd candidates

(c) given “JESSE”

Jesse Jackson, 2nd candidate, and Jesse Jackson Jr., 3rd candidate

(d) given “NOMO”

Hideo Nomo, pitcher of L.A. Dodgers, 2nd candidate

(e) given “LEWIS”

Lewis Schiliro, FBI, 2nd candidate

Figure 16: Name-to-Face Retrieval

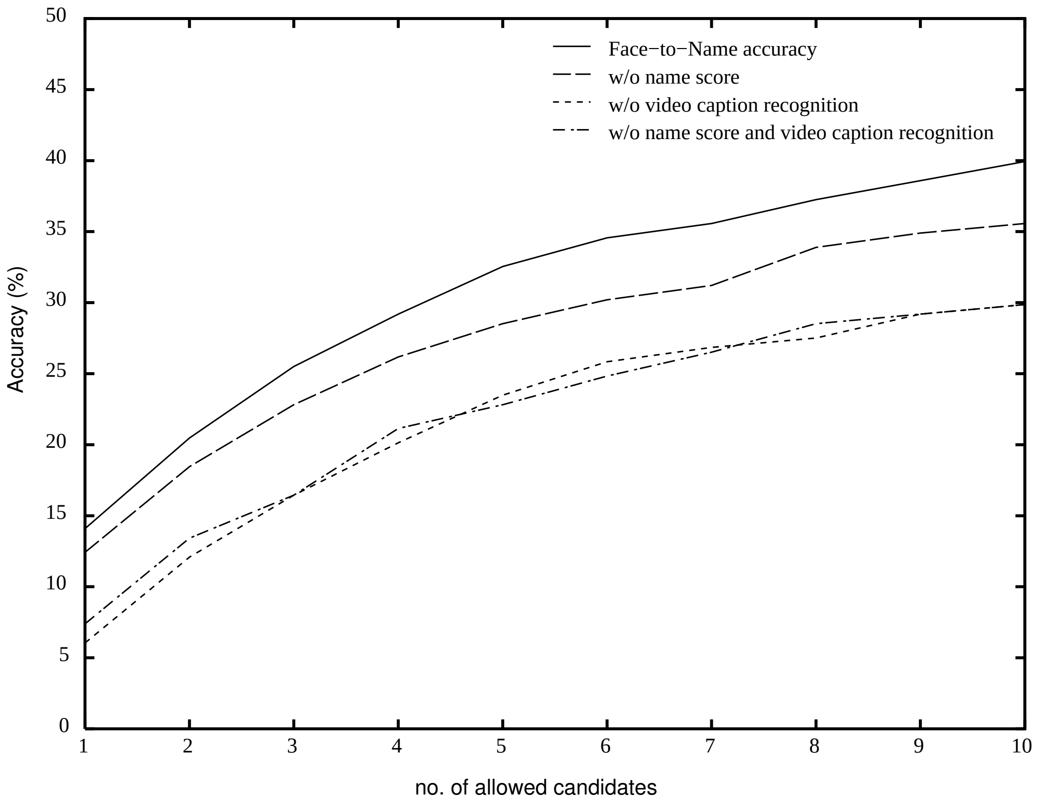


Figure 17: Accuracy of Face-to-Name Retrieval

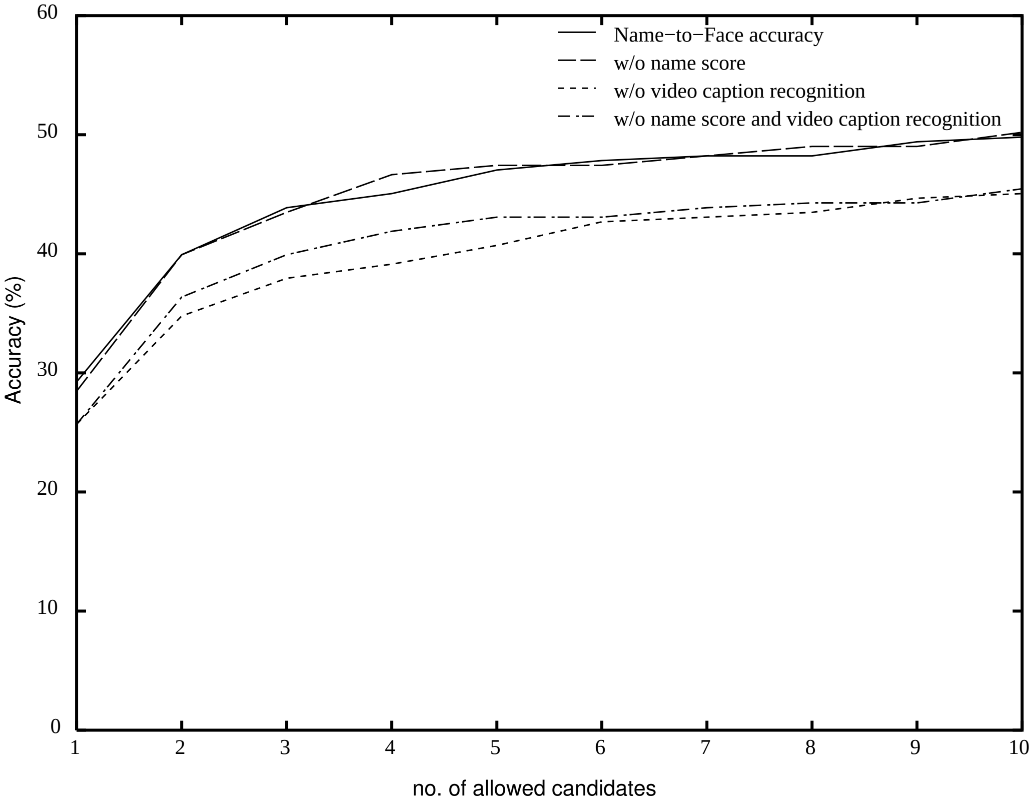


Figure 18: Accuracy of Name-to-Face Retrieval