2

Recognition of Characters from Streaming Videos

Tanushyam Chattopadhyay,
Arpan Pal and Aniruddha Sinha
Innovation Lab, Kolkata,
Tata Consultancy Services Ltd.
India

1. Introduction

Over the past few years, Video has become one of the prime source for recreation, be it Television or Internet. Television brings a whole lot of professionally produced video content (International or local, sports or educational, news or entertainment) to the home for the masses. Similarly, Internet hosts a whole lot of video content uploaded by other users. Understanding the context of the video automatically can open up avenue for a lot of value-added applications. Now, what do we mean by understanding the context? Video context is usually associated with audio, image, graph, text etc. that are embedded within the Video – these are the information that help us understand the content of the video. Video texts can provide a lot of contextual information on the video clips.

In general, there are two types of texts embedded inside video namely, scene texts and artificial texts. Scene texts appear naturally in scenes shot by the cameras. Artificial texts are separately added to video frames (normally in Production Studios) to supplement the visual and audio contents (Lienhart, 1996). Since artificial text is purposefully added, it is usually more structured and closely related to context than a scene text.

The text data in video frames contain useful information for automatic annotation, indexing and summarization of the visual information. Extraction of the text information involves the following processes –

1. Detection of Text Region
2. Localization of Text Region from the detected region
3. Tracking of Text from Localized Region
4. Extraction of Tracked Text
5. Enhancement of the Extracted Text
6. Recognition of the text from the Enhanced Input
7. Post-processing (language dependant) of Recognized Text

This is elaborated in Fig. 1.

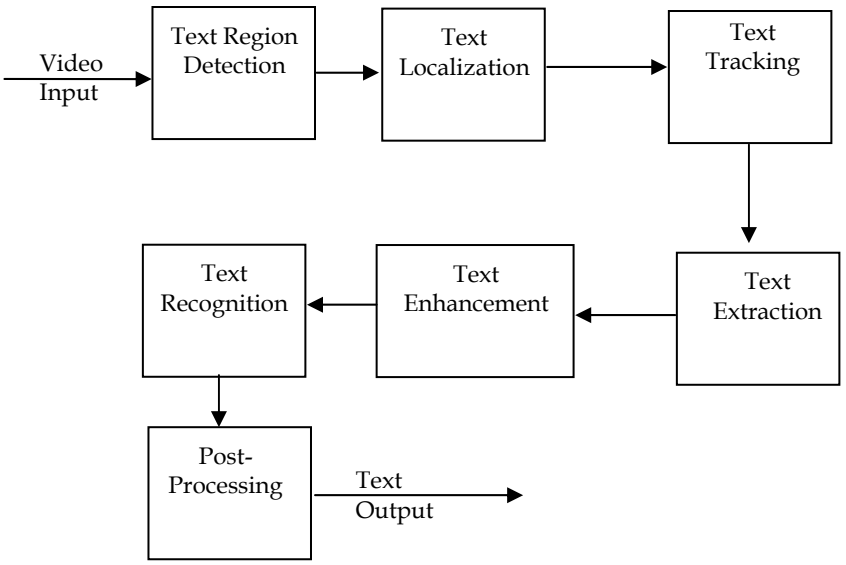


Fig. 1. Flow Diagram of the Character Recognition Process

There are several challenges in the Text Information Extraction from live videos (as compared to standard OCR problem of scanned documents) –

1. Very Low text resolution often affects the reliability of the available Optical Character Recognition (OCR)
2. Text is often embedded on complex background so text separation from background is a difficult task.
3. Text has varying size and font style
4. Text may have touching characters
5. Video has changing contrast and brightness
6. Quality of the Video Signal can vary depending upon the source (Source can be stored media, video broadcast over satellite or cable, video downloaded or streamed from internet etc.)
7. Real-time requirement for extracting text on-the-fly as the Video plays
8. Moving text in the video in form of Tickers (stock, weather, breaking news etc.) poses a challenging Text Tracking problem

To address these challenges, one needs good and robust pre-processing techniques for enhancing the text before passing to standard OCR. Even after good amount of pre-processing, there is no OCR which can give 100% accurate results in the presence of uncontrolled environment. Hence there is a need for post-processing of the OCR output which is then corrected using the help of dictionary, language model and natural language processing in the presence of the extracted context.

Section 2 talks about the Text Localization problem. It first outlines the different formats of the video that can be available as inputs. It then discusses about three different techniques for text localizing -

- Pixel Domain Processing
- Compressed Domain Processing
- Hybrid processing in both domain

Section 3 talks about the Text Tracking problem. Once the candidate text regions are localized, the very next activity should be confirming them as text regions. It can be done by utilizing the fact that text regions would usually be slowly changing compared to video. Two techniques of text tracking are discussed –

- Motion-vector based processing in the compressed domain
- Block Matching method

Section 4 discusses about Video Screen Layout Segmentation problem. Once the text regions are identified, the layout of the video screen is marked. The video background / foreground analysis is done to identify scene texts and artificial texts so that these are marked as separate text segments.

Section 5 talks about Text Pre-processing techniques in the context of video. It mainly talks about three techniques –

- Text Enhancement
- Binarization
- Touching Character Segmentation

Section 6 describes the main Optical Character Recognition techniques. In addition to describing the OCR theory, it also introduces two popular Open-source OCR technologies available – GOCR (GNU Optical Character Recognition) and Tesseract OCR, and gives a comparative analysis of the two.

Section 7 discusses about the Text Post-processing techniques. In particular, it covers three areas –

- Natural Language Processing (NLP) based Spelling Correction
- Dictionary based Spelling Correction
- Language Model based Word Correction

Finally section 8 lists a few applications from real-life that can use Video OCR technique to create compelling use cases for the end user.

2. Text Localization from the video

The format of the video which is the input for the proposed recognition system may be different for different source of origin. The input video may come from a Direct To Home (DTH) service or in form of Radio Frequency (RF) cable. In case DTH, the input video is in MPEG or any other compressed digital video format and on the other hand in case of RF cable feed video, the input is an analog video signal. In the second case initially the video is digitized for further processing. The Text Information Extraction (TIE) module localizes the candidate text regions from the video. The state of the art shows that the approaches for TIE can be classified broadly in two sets of solution: (i) Using pixel domain information when the input video is in raw format and (ii) Using the compressed domain information when the input video is in compressed format. We shall give a brief overview of both type of

approaches as the input to the system can also be either raw or compressed depending on the service provider namely RF cable or DTH respectively. A comprehensive survey on TIE is described in the paper (Jung 2004) where all different techniques in the literature between 1994 and 2004 have been discussed. So we shall discuss about the work between 2005 and 2010.

2.1 Using the pixel domain video information

The (Jung, 2004) paper classifies the pixel domain TIE in two classes namely (i) Region based (RB) and (ii) Texture based (TB) approach. TB approach can be classified into two classes namely (i) Connected component based (CC) and (ii) Edge based (EB) approach. But most of the recent works are based on texture based approach or edge based approach. It is also observed that the authors prefer to use different features followed by a classifier to localize text regions. A comprehensive survey on the recent work is presented below.

TB approach: In (Lee 2007) 12 wavelet based features are given as input to some classifier to recognize TIE. In (Shivakumara, 2009) a gradient difference based texture is used to localize the graphics and scene text from the video. Moreover they have applied a novel zero crossing technique that outperforms the projection profile based technique to identify the bounding boxes. In (Shivakumara, ICDAR 2009) authors have used wavelet transform, statistical features and central moments for detection of both graphics and scene text. They have used projection profile and heuristics to reduce the false positives. In (Emmanouilidis, 2009) structure elements are used to localize texts from binarized printed documents. (Y. Jun, 2009) have used local binary patterns to extract text features followed by a polynomial neural network (PNN) for classification. (J. Zhong, 2009) have used a sliding window over the frame to extract hybrid features followed by a SVM classifier to localize text regions. They have used morphological operations and vote mechanism to reduce false positives.

EB approaches: (Ngo, 2005) proposed an approach based on background complexities. They presented an iterative method to remove non textual regions based on edge densities. (M. Anthimopoulos, 2008) have used edge based features with minimum computational complexity to localize the text regions so that there is no false negative. They have used SVM to minimize the false positives in turn. (P. Shivakumara et.al., ICME 2009), (P. Shivakumara, IAPR 2008), (P. Shivakumara, ICPR 2008) have used edge based features and heuristic based filters to localize text regions and applied some geometry based heuristics to reduce the false positives.

Combined feature based approach: (Y. Song, 2009) have used degree of texture rough-detail to localize the candidate text regions and subsequently used Sobel edge operator to minimize the false positives. (S. Yuting, 2008) have used both texture and edge based approach to localize the candidate text regions and then they have applied SVM to segregate background from foreground.

2.2 Using the Compressed domain video information

Though there has been a lot of work done on pixel domain TIE, there is only a little amount of work can be found on TIE when the input is a compressed video. (Gargi et.al., 1999) and (Lim et.al., 2000) use DCT coefficients of I frames and number of Macroblocks (MB) decoded as Intra in a P or B frame for TIE. (Zhong et. al., 1999) uses the horizontal and vertical DCT texture to localize TIE and refine the candidate text regions respectively. (Qian and Liu,

2006) use DCT texture of compressed video to detect TIE. They have considered diagonal edges to handle the Asian Languages like Chinese, Japanese which are not taken care in other papers. They have also used Foreground (FG) background (BG) integrated video text segmentation method to make it robust even in case of complex BG. (Zhong et. al., 2000) also used horizontal and vertical energy computed from the DCT coefficients of MPEG stream for TIE. They have used Morphological operations in post processing. (Xu et al., 2009) used DCT texture followed by a 3x3 median filter in spatial domain for TIE. They have also used some heuristic that the text must reside for at least 2 sec to remove the false positives.
Now the state of the art clearly reveals that a huge amount of research is going on to detect text regions from a streamed video. But the problem with the existing solutions is that they are not considering H.264 as an input video format which is coming to market as a future generation video Codec. Moreover the compressed domain features are more or less based on the texture property of the video. They didn't consider the edge based features that gives good result in pixel domain. Over and above the accuracy of compressed domain approaches are not as accurate as those obtained from pixel domain approach. But the novelty of the compress domain TIE is that they are computationally very efficient. So we have proposed an approach where the text regions are initially localized using compressed domain features of H.264 video and then they are further processed in pixel domain to remove the false positive. As the compressed domain TIE eliminates a huge part of the video at very beginning of the processing, the complexity of the proposed system is also within acceptable range. We are describing the method in brief below.

2.3 Text Localization using Hybrid Approach

2.3.1 Localization of the text region

The input to the proposed system is video. Now the video format may be different for different scenarios. In this paper we have considered all those cases.
In the first case, we assume that the input is coming from DTH and thus the video is in MPEG format. So in this case we have used the text localization module based on the work described by Jain et al.
In the second case we assume that the input is coming from RF cable. In that case we use an AD converter and use the raw video data as input.
In case of PVR enabled boxes where the video can be stored in H.264 format; one H.264 encoder converts the raw input video into compressed H.264 format.
Our proposed methodology is based on the following assumptions derived from observations.

- Text regions have a high contrast
- Texts are aligned horizontally
- Texts have a strong vertical edge with background
- Texts of Breaking news persists in the video for at least 1-2 seconds

Our proposed method is based on the following two features which can be directly obtained from compressed domain H.264 stream during decoding it.

2.3.2 DC component of transformed Luma coefficient based features

In H.264 4x4 Integer transformation is used which is different from the 8x8 DCT transformation of MPEG series video codec.

DC components of the integer transformed luma coefficients is a representative of the original video at a lower resolution. In H.264, unlike previous video codecs, 4x4 block size is used. So it gives the iconic representation of the video more precisely.
The pseudo code of the algorithm in the proposed method is given below:

- Get the Luma DC value (dc_l) for each 4x4 sub block from decoder
- Compute the first order difference ($\partial_x(dc_l)$ and $\partial_y(dc_l)$) of dc_l with neighbouring sub blocks in x and y direction.
- From observation it is found that the difference is very high for a high contrast region. Obtain such $\partial_x(dc_l)$ and $\partial_y(dc_l)$ for different sub-blocks (not including the test sequences)
- Run K-Means algorithm (with K = 2) on them and find the centroid (τ_{dc}) for the high valued cluster.
- If $\partial_x(dc_l)$ or $\partial_y(dc_l)$ is greater than the experimentally obtained threshold value (τ_{dc}) mark that MB as a candidate text in this frame and store this MB number in an array (a_l)

2.3.3 De-blocking filter based feature

Based on our observations on different TV shows containing texts, it is found that the texts are displayed with high contrast difference from the background. As a consequence the texts results in a strong edge at the boundaries of text and background. But in this approach an additional time complexity is required for detecting edges. One of the new features of H.264 which was not there in any previous video CODEC is deblocking (DB) filter. We have used the edge information extracted from the decoder during the process of decoding without computing the edges explicitly.
A conditional filtering is applied to all 4x4 luma block edges of a picture, except for edges at the boundary of the picture and any edges for which the DB filter is disabled by disable_deblocking_filter_idc, as specified in the slice header. This filtering process is performed on a macroblock after the picture construction process prior to DB filter process for the entire decoded picture, with all macroblocks in a picture processed in order of increasing macroblock indices. For each macroblock and each component, vertical edges are filtered first. The process starts with the edge on the left-hand side of the macroblock proceeding through the edges towards the right-hand side of the macroblock in their geometrical order. Then horizontal edges are filtered, starting with the edge on the top of the macroblock proceeding through the edges towards the bottom of the macroblock in their geometrical order.
The pseudo code for selecting candidate frames using this feature is given below:

- Get the strength of DB filter for each MB
- If it is a strong vertical edge Mark that MB as a candidate one

2.3.4 Identifying the Text regions

The pseudo code for removing the non textual part is as bellow:

- For each candidate MB, identify the X and Y coordinate top left position for each MB (c_x and c_y)
- Find the frequency (f_r) of candidate MB in each row.
- Remove all MBs from a_l If f_r <2
- Check for continuity of MBs in each row: For this check the column number (c_x) for candidate MBs in a row.
- If $c_x(i+1) - c_x(i) > 2$ unmark the MB from a_l Where $c_x(i)$ is the column number for i^{th} candidate MB in a particular row
- To ensure that time domain filtering we store one frame into buffer and display the i^{th} frame while decoding the $(i-1)^{th}$ frame.
- Unmark all candidate MBs in i^{th} frame if there is no candidate MB in adjacent $(i-1)^{th}$ frame and $(i+1)^{th}$ frame.
- Finally all marked candidates MBs are decided as Text content in the video.

2.3.5 Morphological closing

Because of the video quality, in V_{cont} the text components are not getting disjoint. Moreover the non textual regions are also coming as noises. So a morphological closing operation is applied on V_{cont} to get a video frame V_{morph}

$$V_{morph} = Dilate(Erode(V_{cont}))$$

In this application we have used a rectangular structural element with dimension of 3x5.

2.3.6 Confirmation of the Text regions using shape feature

This is the method to remove the non textual regions from the video frame based on the observation that text characters are adjacent. The pseudo code for removing the non textual part is as below:

- Run connected component analysis for all $P_c \in V_{morph}$ to split the candidate pixels into n number of components (c_i) where P_c is a pixel in V_{morph}
- Find the area for each c_i
- Remove the components with area smaller or greater than two experimentally obtained threshold values.
- Remove all components for which compactness $compactness > 1.0 \; or \; compactness < 0.2$ where $compactness = PixelCount / Area$

- Find the mode for x and y coordinate of top left and bottom right coordinate (tl_x, tl_y, br_x, br_y) for all remaining components
- Find the threshold (τ) for removing non textual components as
- $\tau = \text{mod}\, e(median(pos_i) - pos_i)$

  Mark all c_i for which $median(pos_i) - pos_i < \tau$ to get a video frame containing only candidate text regions (V_{cand})

2.3.7 Confirmation of the Text regions using Temporal Consistency

This portion of the proposed method is based on the observation that texts of Breaking news persists in the video for at least 1-2 seconds. V_{cand} sometimes contains some noises coming because of some high contrast regions in the video frame satisfying all the shape constraints. But these noises usually come for some isolated frames only. In a typical video sequence with 30 FPS one frame gets displayed for 33 millisecond. So, we ruled out those candidate frames to finally obtain a video frame containing only text regions V_{text}.

2.3.8 Decision making process

The image received in .264 format is decoded by H.264 Decoder and is processed to obtain the localized text. This is done using compressed domain features. The text after getting compressed undergoes text localization processing.

The first step involves computation of vertical and horizontal energy of the sub block based on the assumption that the blocks with text have high energy levels. After we get the information about the sub block, we check which rows contain high vertical and horizontal energy which indicate the presence of text. The regions with lower energy are marked as black after they are checked using a threshold value based on the analysis of the different energy levels in a row. Then we plot a histogram to show the energy levels in a row. With the help of histogram we determine the two major peaks of energy levels. If the graph is not smooth enough we perform mean filtering and obtain the two major peaks. The mid value of the peaks is taken as threshold and the values of the pixel above threshold is set as white and the values below the threshold is set as black. This step is called binarization. Thus we obtain a localized text with shows the regions where the text is located. The image obtained also contains some false positives i.e. noise along with the text detected. Hence, we go for some morphological operations and filtering which enhance the image and give better localization with less false positives. The image obtained is further processed to represent the text regions as white rectangular blocks so as to mark the presence of text.

3. Text Tracking

Once the candidate text regions are localized, the very next activity should be confirming them as text regions. One way to achieve this in a video is using the temporal consistency of text regions. It is observed that if any text region exists in the video, it usually persists for some consecutive frames. Thus there is a need for tracking module that can track the motion of the text.

The text tracking algorithms described in the literature can be classified into two classes based on the type of input video. In case compressed video file as input, Motion Vector (MV) of the compressed video is used to track textual region (TTR). On the other hand if the video comes as a raw file, the authors have suggested different techniques for motion estimation (ME). Some common ME techniques are Minimum Mean Absolute Difference (MMAD) or Sum of Squared Difference (SSD). But the problem with SSD based ME is that they assume only translational motion of text and thus they are not Affine Transformation Invariant in nature. In this section we shall describe each type of TTR algorithms very briefly.

MV based approach: MV of the compressed MPEG video file is used for TTR in the works by (Antani et. al., 1999), (Gllavata et al., 2004) and (Gargi et al., 1999). Here is a brief overview of the approach described by (Gllavata et al., 2004).

They have used normalized motion vector which is derived by dividing the actual motion vector (both x and y component) by the frame distance. Frame distance is the difference between current and reference frame. The tracking is done only if the frame is a frame of type B or P. The new position is computed by adding the mode motion vector of the block where the text resides. Applying motion vectors for text tracking is difficult due to noise factors and the problem of identifying which motion vectors probably describe the text motion and which the background motion.

Block matching methods: Two major block matching techniques are MMAD and SSD.

MMAD: In this method for each block in the current frame the following operations are performed:

- Compute the absolute value of difference (AD) between pixels values of current frame and the reference frame.
- Compute the average of the AD and let it be denoted as MAD
- Compute MAD for all neighbouring blocks within the search window
- Find the minimum of all the MAD and that block with minimum MAD is marked as the candidate matching location
- The euclidian distance in X and Y direction are derived as the distance between them

4. Video Screen Layout Segmentation

Once the candidate text regions are localized, and tracked, the false positives are mostly removed and the candidate text regions are obtained. But within streamed video texts are usually inscribed in different areas. For example in the video frame shown in Fig. 2, eight different regions (R1-R8) can be found containg text information. Now for text in R1 and R5 represents some Breaking news, R2 is a text within a graphics, R3 and R4 gives information about the reporter and reporting venue, R7 gives the details of breaking news, R6 gives the channel logo and text and R8 gives the ticker news.

Now the video page layout segmentation module plays an important role for getting better performance in term of time complexity and as well as for recognition accuracy. For example R5, R6 and R7 are very contiguous text lines and so they are given as a single entry for OCR. Now In R5 text color (FG) is black and background color (BG) is yellow. On the other hand in R7 FG is yellow and BG is black. So the binarization module discards either of the FG as BG when given to OCR. Moreover R6 does not contain any relevant information

showing in R5 or R7. So there arises a need for good Video Page Layout segmentation module.
Document page layout segmentation is a very common problem for Document Image Analysis. But the text document layouts are much simple than those of actual videos.

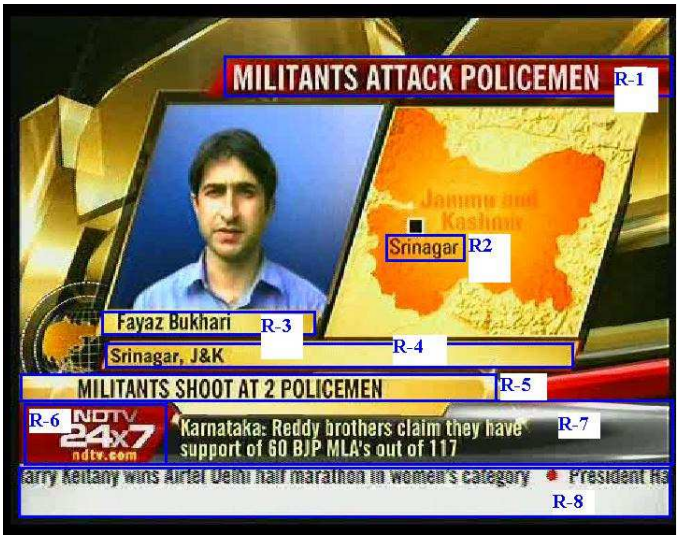


Fig. 2. Presence of text in different segments

We have used a version of XY tree cut method for document page layout segmentation which is based on quantized color space.

5. Text Recognition

5.1 Pre-processing

Once the ROI is defined manually we can directly give this ROI to the recognition module of some OCR engine. But it is found that a lot of blurring and artifacts in the ROI reduces the recognition rate of the OCR. We have used two different algorithms to condition the image before giving it to the input of the OCR.

Interpolating the image to higher resolution and applying Low pass Filter (LPF):

In this method we have done the following steps:

- Apply six tap FIR filter with filter coefficients (1, -5, 20, 20, -5,1) to zoom the ROI two times in height and width
- Apply simple interpolation technique to zoom it further two times in height and width
- Apply DCT on the higher resolution image
- Apply Butter Worth Low Pass Filter to discard the high frequency components
- Apply Inverse DCT to reconstruct the image in higher resolution
- ICA based approach can also produce very good result.

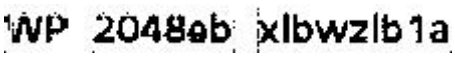
5.2 Binarization

The output of the preprocessed model is then binarized using an adaptive thresholding algorithm. There are several ways to achieve binarization so that the foreground and the background can be separated. However, as both the characters present in the relevant text region as well as the background are not of a fixed gray level value, adaptive thresholding is used in this approach for binarization. To obtain the threshold image, Otsu's method is used in this solution.

(a)

(b)

(c)



(d)

Fig. 3. (a) Original Video (b) After Preprocessing (c) After Binarization (d) After Touching Character segmentation and rescaling

5.3 Touching Character Segmentation

Once the binarized image is obtained very frequently it is observed that the image consists of a number of touching characters. These touching characters degrade the accuracy rate of the OCR. Hence the Touching Character segmentation is required to improve the performance of the OCR. Here is the pseudo code for the same

- Find the width of each character. It is assumed that each connected component with a significant width is a character. Let the character width for the i^{th} component be WC_i
- Find average character width $\mu_{WC} = {}^{1}\!/\!_{n}\sum_{i=1}^{n} WC_i$ where n is the number of character in the ROI
- Find the Standard Deviation of Character Width (σ_{WC}) as $\sigma_{WC} = STDEV(WC_i)$

- Define the threshold of Character Length (T_{WC}) as $T_{WC} = \mu_{WC} + 3\sigma_{WC}$
- If $WC_i > T_{WC}$ mark the i^{th} character as candidate touching character
- The number of touches in i^{th} candidate component is computed as $n_i = \left\lceil \frac{WC_i}{T_{WC}} \right\rceil + 1$
- Divide WC_i in n_i equally spaced segments

Results of each of the above steps are depicted in Fig. 3.

6. Optical Character Recognition

The pre-processed output is fed to standard OCR engines for recognizing the characters. In a video frame there can be texts with different font sizes, different styles, varying intensities with complex background. The adverse effects due to these are taken care in the previous section by pre-processing, where the binarized output is used for the subsequence character recognition operations.

A standard OCR engine consists of the following three stages:

a. Segmentation and creation of bounding box and normalization
b. Feature extraction
c. Classification

A set of training and test data is used to build an OCR. Initially these data is scaled to normalize the size and create a bounding box around each character. The images of the normalized characters are processed to generate characteristics vectors of features that are further used to classify the characters. Some of the basics features used in the OCR analysis are compactness, anisometry, zero crossing etc.

The standard OCR systems are usually used as black-box and don't allow the user for tuning or re-training which leads to a drastic reduction in performance, especially whenever the input to the OCR is not a regular scanned document.

A couple of examples of public domain OCRs are GOCR and Tesseract. These are primarily meant for recognizing texts from the scanned documents. We compare the performance of these OCR engines with and without the pre-processing algorithms. Fig. 4 (a)-(j) shows different images where the text ROI are extracted using text localization and video layout segmentation. These images are then binarized and fed to OCR engines to extract the texts.

A comparison of the accuracy of the extracted texts is shown in Table 1, where it is seen that Tesseract clearly outperforms GOCR. For the image (b), GOCR fails to recognize a single character, whereas Tesseract is able to recognize two words ("SMS" and "56633") correctly. Similar is the case for the image (j) where none of the characters are recognized by GOCR and all the words are recognized by Tesseract.

Further improvement of recognized text is achieved by using the proposed pre-processing algorithms. Table 1 clearly demonstrates the marked improvement of using the pre-processing for the GOCR engine. However, the Tesseract performs much better compared to GOCR, hence we hereby analyze the improvement results obtained with the Tesseract. For image (a), the name "Reema Sen" is recognized correctly after the pre-processing, whereas

the names "Govinda" and "Rajpal" failed by one character each. In case of image (b), there is no improvement in the recognized words but it doesn't degrade the performance in recognizing the key information ("SMS" and "56633"). The preposition "to" in image (c) is recognized correctly. In case of small words, the pre-processing clearly improves the performance due to the filtering algorithm in the pre-processing.

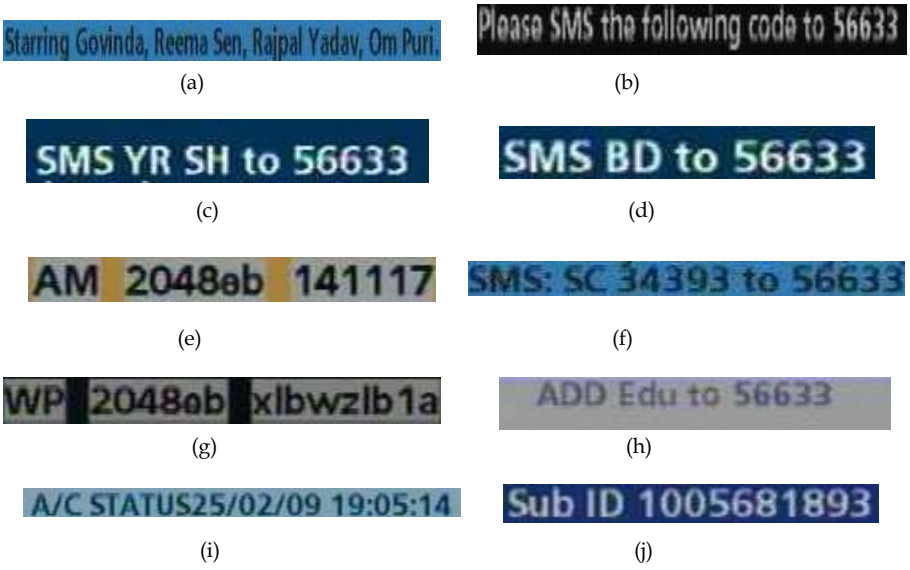


Fig. 4. (a) – (j) Represents different images under consideration.

Image	Output of GOCR	Output of Tesseract	After Applying Proposed Pre-processing Algorithms	
			GOCR	Tesseract
(a)	Sta_ring Govind_. Reem_ _n. RajpaI Yadav. Om Puri.	Starring Guvinda, Rcema Sen, Raipal Yadav, Om Puri.	Starring Govind_. Reem_ _n. RajpaI Yadav. Om Puri.	Starring Guvinda. Reema Sen, Raipal Yadav. Om Puri.
(b)	_____ ___ ___ _____ ____ __ __	Pluww SMS thu fnlluwmg (adn In 56633	___ SMS th_ folIcmng cod_ to S__	Planta SMS tha Iullmmng mda tn 56633
(c)	SmS YR SH to	SMS YR SH in 56633	SmS YR SH to _____	SMS YR SH to 56533
(d)	_m_ BD to _____	SMS BD to 56633	SMS BD to S____	SMS BD to 56633
(e)	AM t___o_,_b _q____	AM 2048eb 141117	AM tOa_gb _q____	AM 2048eb 141117

(f)	_M_= _ _A___ to Sd___	SMS: SC 34393 tn 56533	_M_= _ _A___ to Sd___	SMS: SC34393 tn 56633
(g)	_W _ ' _b _ Ib_lb _a	W6.} 048abl;lbwzIb1a	___ __Y_b yIbw_Ib_a	WP 2048ab Mlbwzlb 1 a
(h)	ADD Ed_J to S____	ADD Eau to $6633	ADD Ed_J to S____	ADD Edu to 56633
(i)	AIC STAlUSlS/OUO_ t_;OS;t_	AIC STATUS25/02/09 1 9:05:1 4	mlC S_ATUSlS/OUO_ t_;OS=tA	A/C STATUS 25/02/09 1 9:05:14
(j)	_ _____'__	Sub ID 1005681893	WbID_OOS_B_B__	Sub ID 1005681893

Table 1. Output of the different OCR engines before and after applying the image pre-processing algorithms

It can be seen from Table 1 that even after the improvement due to pre-processing algorithms, we don't achieve complete recognition of the texts. Thus there is a need for context based post-processing of the OCR output. The output of image (a) can be corrected by using a context based dictionary database of proper nouns, whereas the output of image (b) can be only be corrected using certain high level information and Natural Language Processing (NLP). The preposition error in image (f) can be corrected using Language Models (LM).

7. Text Post-Processing

The output of commercially available OCRs is pretty acceptable for neatly scanned document images with nearly 300 dots per inch resolution. However, in case of streamed videos, the video quality is comparatively very poor; as a consequence the accuracy of OCR is also not very good as seen in Table 1. Hence there is a need for some post processing module like language module based approach, NLP based approach or dictionary based approach to rectify the errors and increase the recognition accuracy. There has been a considerable amount of research done on correcting the words automatically in the output of OCR. A good survey of the research in this area is provided by (Kukich, 1992).

7.1 NLP Based Spelling Correction

Natural language processing is a computer based approach in analyzing and representing texts using a range of computational techniques for achieving human-like language processing. The goals of NLP are primarily to

a. Parse and paraphrase a text
b. Translate to another language
c. Respond to the queries about the text contents
d. Infer from texts

However, the tools of NLP can be used to perform the spelling and word correction using the knowledge of different levels of a language which are also used by humans to gain understanding (Liddy, 2003). The capability of NLP systems too depend on the utilization of the following levels of language:

a. Phonology – This is a study which deals with the inventory and interactions of sounds (Phonetics) in a specific language. This forms the basis for further work in morphology, syntax, discourse etc.
b. Morphology – This deals with the study of formation and structure of words in a language.
c. Lexical – This refers to classes of things or to concepts in a language.
d. Syntactic – This provides the principles and rules for constructing sentences in languages.
e. Semantic – This is a study of meaning in a language.
f. Discourse – This deals with the study of deriving meaning from the connections of multiple sentences.
g. Pragmatic – This is a study which deals with how context contributes to meaning in a language.

Most of the NLPs use the lower levels of processing as these are thoroughly researched and implemented; lower levels deal with smaller units of analysis e.g., morphemes, words and sentences. Some of the major applications of NLP are information retrieval, information extraction, question-answering, summarization and machine translation and dialogue systems. The word level error correction capability of NLP is a powerful tool to improve the accuracy of text extraction compared to a standalone OCR. This can be used to correct articles, prepositions and verbs.

(Pal et al., 2000) proposed OCR error correction in morphologically rich Indian language, where they have shown 84% word correction for a single character error.

(Chodorow et al., 2007) presented a work on detecting errors in preposition for non-native English speakers. They have proposed maximum entropy (ME) model to estimate the probability of prepositions in their local context to detect the errors due to "incorrect selection". The ME combined with Bayesian classifiers improves the precision to 0.88.

Generative probabilistic OCR model is proposed by (Kolak et al., 2003) which is implemented using a weighted finite state model (FSM) framework of AT&T FSM Toolkit. The reduction in word error rate is 70% over the English-on-French for a standard OCR with an overall accuracy of 97%.

(Taghva and Stofsky, 2001) presents the selection of candidate words by using multiple knowledge sources for spelling correction. They proposed a system which has three parts namely, a two level word generator for incorrect words, a confusion generator for longest common subsequence and confusion of words, finally a user interface that allows the user to review the candidate corrections and change accordingly.

Thus we see that NLP plays a very important role in improving the word recognition rate which helps for better understanding of the sentences.

7.2 Dictionary Based Spelling Correction

Spelling correction using dictionary is the most widely used post-processing module for OCR. The most popular commercially available software for the spelling correction in OCR is Abby's Finereader. The video OCR for digital news archives proposed by (Sato et al., 1998) matches the OCR output with the words from the dictionary. A distance measure based on correlation is used to calculate the similarity of the word in OCR output with the dictionary database. They have used two types of dictionary, namely, Oxford Dictionary for general words and database for noun words to accommodate names of people, organization

or places. It is shown that after the dictionary based spelling correction the word recognition rate increased to 65.2% from 48.3%.
(Hauptmann et al., 2002) has demonstrated an increase in accuracy of Information retrieval from broadcast video by using n-gram analysis and dictionary spelling correction on the output of the OCR. The MS word is used to perform the spelling correction. The n-gram post-processing improved the word recognition accuracy by 100% compared to the basic video OCR output.
Sometimes, the post-processing algorithms and the actual OCR are integrated to allow information exchange between the two. (Hanson et al., 1976) reported 2% word error rate and 1% reject rate without using any dictionary.
By using an augmented dictionary and ignoring punctuation, (Sinha and Prasada, 1988) achieved 97% of word recognition with a Viterbi type algorithm.
In the context of text recognition, certain possible way of partitioning a dictionary is proposed by (Sinha, 1990). The word-length, word-envelop and character combination is used for the basis of partitioning the dictionary.
Wherever possible, it is always recommended building a context based dictionary which improves the word recognition rates further compared to a normal dictionary.

7.3 Language Model Based Word Correction

The spelling checkers often used in correcting texts are mostly based on non-word errors. The words which are outside the valid list of words in a particular language are treated as erroneous words. However, there are several types of spelling errors, where the words being part of the language are incorrectly used in the current context. These require recognizing the context and creating a model for the same. *Confusibles* is a class of error, where the words belong to a particular language but are incorrectly used in their local context. Most of the research is done for confusables due to homophony (by, bye or center, centre) and similar spellings (abjure, adjure or founder, flounder). The context sensitive spelling errors are tackled by using a specially trained machine learning classifier for a specific confusable set. Word prediction based on the knowledge of a language is an alternative approach to solve the confusable errors. Language modeling can be used to solve the errors by selecting the best alternative from the confusable sets. Probabilities are assigned to sequences of words in a language model to predict the most likely word in a given context.
Language models used as generic classifiers to build a system that can be generic for all set confusable disambiguation is presented by (Stehouwer et al., 2009).
(Srihari et al., 1983) treats OCR as a black box and performs n-grams analysis on word and/or character level along with character confusion probabilities. They report up to 87% error correction on artificial data while relying on a lexicon for correction.
Several knowledge sources are specific to a particular language can be used to aid the test extraction. (Bansal and Sinha, 2000) proposed integrating several knowledge sources for Devanagari script (for Hindi which is official language of India) in hierarchical manner.
The time dependent language model (TDLM) based on weighted mixture of long-term news scripts and latest scripts as training data is used to improve the correctness of extraction of text information from the broadcast video by (Kobayashi et al., 1998).

Given all the above tools to improve the accuracy of OCR output, one is still far away from achieving 100% accuracy in recognizing the texts especially for the videos generated or captured in an un-controlled environment.

8. Some Real-Life Uses Cases

Character Recognition from Streaming Videos can be used in multitude of applications. In most of the applications, the character recognition technology serves as the means for understanding the context of the Video. Once the context of the video is understood, it can be used create a lot of innovative value-added applications. We list out a few such applications in this section.

8.1 Automatic Classification of Videos for Storage / Retrieval

There are different kinds of programs that are broadcast on TV – News, Sports, Entertainment / Movies etc. It would be of immense help if these videos can be classified automatically based on the content. Possible use cases may include offline recording of TV programs for later analysis and Digital Video Recording (DVR). While each of these videos may have certain underlying properties (Sports Videos are more fast-changing, News Videos contain more static scenes etc.), it is really very difficult to classify the videos reliably in that way. A far more interesting and useful approach lie in trying to detect the texts that are coming embedded inside such videos. Once the texts are detected, the very semantics of the text can give enough clues on the type of the Video. For example detection of a "Breaking News" or Stock Tickers would characterize a News Channel (Fig. 5). Detection of "Score" type text can signify a Sports channel (Fig. 5). Presence of sub-titles can signify a movie channel. Once the videos are classified, it becomes easy to store them with the classification. This in turn results in a faster and easier retrieval mechanism.

Fig. 5. TV screen shots depicting "Stock Ticker" in news channel and "Score" in News Channel

8.2 Advertisement Removal for News Video / Movies

Advertisements can also be characterized by the text embedded in them (Fig. 6). Once they are detected, they can easily be removed during storage. This use case is just an extension of that described in section 8.1, however the end-use objective is slightly different. It should be

noted that in order to implement the system in both 8.1 and 8.2, text detection may not be the only technology to be employed – a multi-modal approach of Video Analytics, Audio Analytics and Text Recognition together can provide far better results.

Fig. 6. TV screen shots depicting Advertisement

8.3 Automated Video Indexing for Internet Search

Currently Video Indexing for Search is normally based on File name based indexing. However, texts embedded in the videos can provide far more information about the context of the video. If text recognition engine is run across the complete video, it can yield a set of keywords that best describes the context of the Video (Fig. 7). The complete Search Indexing can be based on these Keywords. Again, this may not be an independent technology, but a complementary one that works hand-in-hand with other Video Indexing Technologies employed to create far more accurate multi-modal implementations.

Fig. 7. TV Video depicting Keyword Texts

8.4 Duplicate News Story Detection

News broadcasts are often accompanied by text tickers that describe the corresponding news event in the video. In applications where video recording is required for multiple TV channels, one can identify duplicate News Stories occurring in multiple channels / repeat broadcasts by just recognizing the accompanying text and comparing the recognized text for

duplication (Fig. 8). This is way one can avoid the duplication in recording and save storage space.

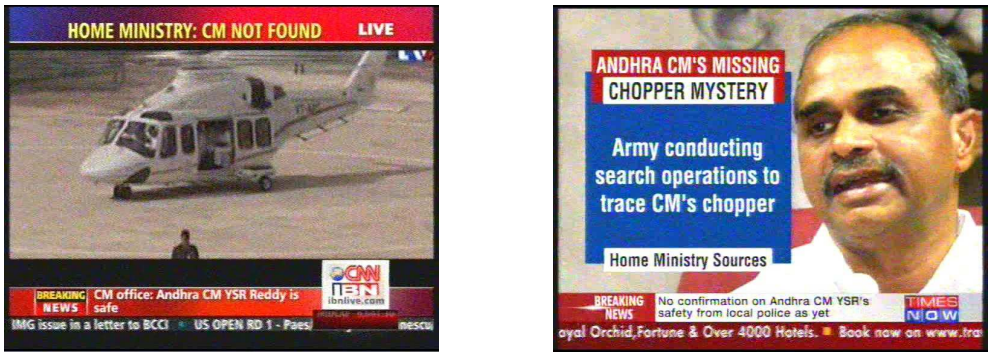


Fig. 8. TV screen shots depicting same texts in two different channel

8.5 Personalized Stock Market Ticker

Normally Stock tickers are embedded into TV broadcast as a continuous stream consisting of all stocks. However, a user may have interest in only a few particular stocks. OCR can be employed in real-time on the stock ticker section of the TV broadcast video and only the stocks of interest can be picked. The picked stock names and their prices can be shown as a separate text overlay ticker on top of the current TV Video (Fig. 9). The Set top Box giving feed to the Television needs to have overlay feature in order to enable this functionality.

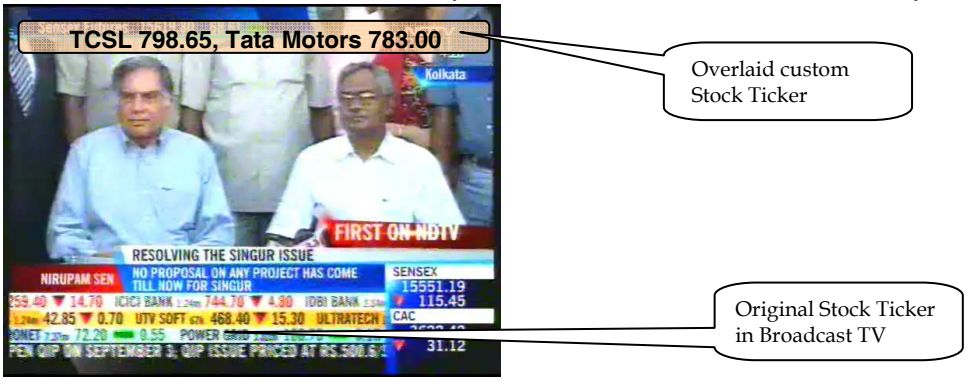


Fig. 9. Intelligent Stock Ticker Overlay on TV

8.6 Personalized Mash-up of Internet News with TV News

As an extension of the same technology described in section 8.5, TV news broadcast that normally has a lot of "Breaking News" text tickers can be a candidate for applying OCR. Once the OCR recognizes the news texts, it can be converted to a set of keywords based on a pre-defined dictionary. These keywords can then be used to fetch related news from Internet by subscribing to different Internet Newsfeed channels. This results in a stream of

news information that is contextually related to the news video currently broadcast on TV. The whole process is depicted in Fig. 10. This news information stream can either be overlaid on top of current TV video or can be stored inside the Set top Box for later access by the user.

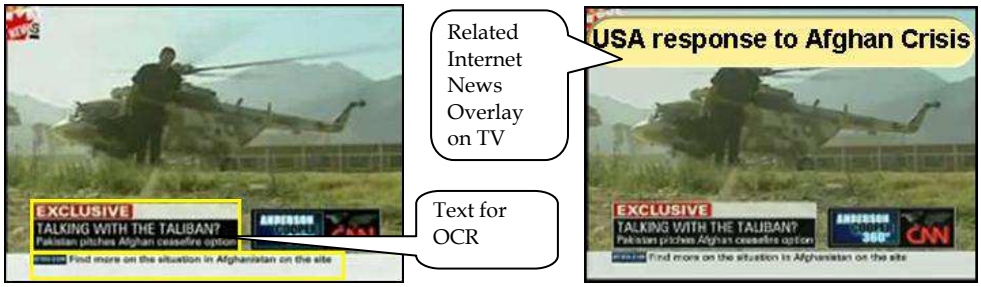


Fig. 10. Intelligent Mash-up of TV News with Internet News

9. References

Alexander G. Hauptmann, Rong Jin, Tobun Dorbin Ng. (2002). Multi-modal Information Retrieval from Broadcast Video using OCR and Speech Recognition, *Joint Conference on Digital Libraries (JCDL '02)*, Portland, OR, pp. 376, July 13-17, 2002

Allen R. Hanson, Edward M. Riseman, and Edward G. Fisher. (1976). Context in word recognition, *Pattern Recognition*, 8:33–45, 1976

C. Emmanouilidis, C. Batsalas, N. Papamarkos. (2009). Development and Evaluation of Text Localization Techniques Based on Structural Texture Features and Neural Classifiers, *Proceedings of 10th International Conference on Document Analysis and Recognition,* pp.1270-1274, 26-29 July 2009

C.-W. Ngo, C.-K. Chan. (2005). Video text detection and segmentation for optical character recognition," Multimedia Systems, vol. 10, No. 3, pp. 261-272, Mar. 2005

Herman Stehouwer, Menno van Zaanen. (2009). Language models for contextual error detection and correction, *Proceedings of the EACL 2009 Workshop on Computational Linguistic Aspects of Grammatical Inference*, pages 41–48, Athens, Greece, 30 March

J. Zhong, W. Jian; S. Yu-Ting. (2009). Text detection in video frames using hybrid features, *Proceedings of International Conference on Machine Learning and Cybernetics*, pp.318-322, 12-15 July 2009

J. Gllavata, R. Ewerth, B. Freisleben. (2004). Tracking Text in MPEG Videos, *Proc. Of ACM*, 2004

Jiangbo Xu; Xiuhua Jiang; Yuxia Wang. (2009). Caption Text Extraction Using DCT Feature in MPEG Compressed Video, *Proceedings of WRI World Congress on Computer Science and Information Engineering,* pp. 431-434, March 31 2009-April 2 2009

K. Jung, K. I. Kim, and A. K. Jain. (2004). Text Information Extraction in Images and Video: A Survey, *Pattern Recognition*, Volume 37, Issue 5, May 2004, Pages 977-997

Karen Kukich. (1992). Techniques for automatically correcting words in text, *ACM Computing Surveys*, 24(4):377–439, December 1992

Kazem Taghva and Eric Stofsky. (2001). OCRSpell: an interactive spelling correction system for OCR errors in text. *IJDAR*, 3(3):125–137, 2001

Kobayashi, Akio Onoe, Kazuo Imai, Toru Ando, Akio. (1998). Time dependent language model for broadcast news transcription and its post-correction, *ICSLP-1998*, paper 0973

Liddy E. D. (2003). Natural Language Processing. In: *Encyclopedia of Library and Information Science*, Editor: Miriam Drake, New York: Marcel Dekker Inc., ISBN: 978-0-8247-2075-9 (hardback) 978-0-8247-2071-1 (electronic)

M. Anthimopoulos, B. Gatos, I. Pratikakis. (2008). A Hybrid System for Text Detection in Video Frames, *Proceedings of The Eighth IAPR International Workshop on Document Analysis Systems*, pp. 286-292, 16-19 Sept. 2008

Martin Chodorow, Joel R. Tetreault and Na-Rae Han. (2007). Detection of Grammatical Errors Involving Prepositions, *Proceedings of the 4th ACL-SIGSEM Workshop on Prepositions*, pages 25–30, Prague, Czech Republic

Okan Kolak, William Byrne, Philip Resnik. (2003). A Generative Probabilistic OCR Model for NLP Applications, *Proceedings of HLT-NAACL*, Main Papers, pp. 55-62, Edmonton, May-June 2003

P. Shivakumara, Q. P. Trung, L. T. Chew. (2009). A Gradient Difference Based Technique for Video Text Detection, *Proceedings of 10th International Conference on Document Analysis and Recognition,* pp. 156-160, 26-29 July 2009

P. Shivakumara, T.Q. Phan, T. C. Lim. (2008). An Efficient Edge Based Technique for Text Detection in Video Frames, *Proceedings of The Eighth IAPR International Workshop on Document Analysis Systems*, pp. 307-314, 16-19 Sept. 2008

P. Shivakumara, T.Q. Phan, T. C. Lim. (2008). Efficient video text detection using edge features, *Proceedings of 19th International Conference on Pattern Recognition*, pp. 1-4, 8-11 Dec. 2008

P. Shivakumara, T.Q. Phan, T. C. Lim. (2009). Video text detection based on filters and edge features, *Proceedings of IEEE International Conference on Multimedia and Expo*, pp. 514-517, June 28 2009-July 3 2009

P. Shivakumara, T.Q. Phan, T. C. Lim. (2009). A Robust Wavelet Transform Based Technique for Video Text Detection, *Proceedings of 10th International Conference on Document Analysis and Recognition*, pp. 1285-1289, 26-29 July 2009

R. Lienhart. (1996). Automatic text recognition for video indexing, *in Proc. ACM Multimedia* Boston, MA, Nov. 1996, pp. 11-20

R. M. K. Sinha. (1990). On partitioning a dictionary for visual text recognition, *Pattern Recognition*, Vol 23, Issue 5, Pages 497-500, 1990

R. M. K. Sinha and Biendra Prasada. (1988). Visual text recognition through contextual processing, *Pattern Recognition*, 21(5):463–479, 1988

Sargur N. Srihari, Jonathan J. Hull, and Ramesh Choudhari. (1983). Integrating diverse knowledge sources in text recognition, *ACM Transactions on Office Information Systems*, 1(1):68–87, January 1983

S. Yu, W. Wenhong. (2009). Text Localization and Detection for News Video, *Proceedings of Second International Conference on Information and Computing Science,* pp. 98-101, 21-22 May 2009

T. Sato, T. Kanade, E. K. Hughes, and M. A. Smith. (1998). Video OCR for digital news archive, in *Proc. IEEE Workshop Content-Based Access Image Video Database*, 1998, pp. 52-60

U. Gargi, S. Antani, and R. Kasturi. (1998). Indexing text events in digital video databases, *Proceedings of 14th Int. Conf. Pattern Recognition*, pp. 916-918, 1998

U. Pal, P. K. Kundu, and B. B. Chaudhuri. (2000). OCR error correction of an inflectional Indian language using morphological parsing, *Journal of Information Science and Engineering*, 16(6):903–922, November 2000

Veena Bansal and R. M. K. Sinha. (2000). Integrating Knowledge Sources in Devanagari Text Recognition System, *IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans*, Vol. 30, No. 4, July 2000

X. Qian, G. Liu. (2006). Text Detection, Localization and Segmentation in Compressed Videos, *ICASSP*, pp 385-388, 2006

Y. Zhong; H. J. Zhang, A.K. Jain. (1999). Automatic caption localization in compressed video, *Proceedings of International Conference on Image Processing*, pp.96-100, 1999

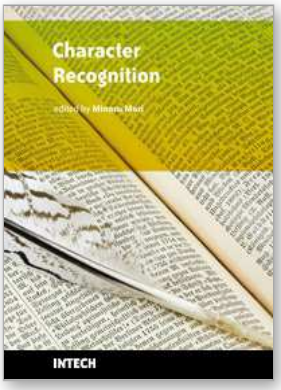
Y. Su, Z. Ji, X. Song, R. Hua. (2008). Caption text location with combined features using SVM, *Proceedings of 11th IEEE International Conference on Communication Technology*, pp.711-714, 10-12 Nov. 2008

Y. Su, Z. Ji, X. Song, R. Hua. (2008). Caption Text Location with Combined Features for News Videos, *Proceedings of International Workshop on Geoscience and Remote Sensing and Education Technology and Training*, pp. 714-718, 21-22 Dec. 2008

Y.-K. Lim, S.-H. Choi, and S.-W. Lee. (2000). Text extraction in MPEG compressed video for content-based indexing, *Proceedings of Int. Conf. on Pattern Recognit.ion*, pp. 409–412, 2000

Y. Zhong, H. Zhang, and A. K. Jain. (2000). Automatic Caption Localization in Compressed Video, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, No. 4, pp. 385-392 Apl. 2000

Y. Jun, H. Lin-Lin, L. H. Xiao. (2009). Neural Network Based Text Detection in Videos Using Local Binary Patterns, *Proceedings of Chinese Conference on Pattern Recognition*, 2009, pp. 1-5, 4-6 Nov. 2009



Character Recognition

Edited by Minoru Mori

ISBN 978-953-307-105-3
Hard cover, 188 pages
Publisher Sciyo
Published online 17, August, 2010
Published in print edition August, 2010

Character recognition is one of the pattern recognition technologies that are most widely used in practical applications. This book presents recent advances that are relevant to character recognition, from technical topics such as image processing, feature extraction or classification, to new applications including human-computer interfaces. The goal of this book is to provide a reference source for academic research and for professionals working in the character recognition field.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821