Journal of Ambient Intelligence and Humanized Computing (2022) 13:1853–1865
https://doi.org/10.1007/s12652-021-02951-1

ORIGINAL RESEARCH

An end to end system for subtitle text extraction from movie videos

Hossam Elshahaby[1] · Mohsen Rashwan[1]

Received: 30 April 2020 / Accepted: 4 February 2021 / Published online: 22 February 2021

Abstract
A new technique for text detection inside a complex graphical background, its extraction, and enhancement to be easily recognized using the optical character recognition (OCR). The technique uses a deep neural network for feature extraction and classifying the text as containing text or not. An error handling and correction (EHC) technique is used to resolve classification errors. A multiple frame integration (MFI) algorithm is introduced to extract the graphical text from its background. Text enhancement is done by adjusting the contrast, minimize noise, and increasing the pixels resolution. A standalone software Component-Off-The-Shelf (COTS) is used to recognize the text characters and qualify the system performance. Generalization for multilingual text is done with the proposed solution. A newly created dataset containing videos with different languages is collected for this purpose to be used as a benchmark. A new HMVGG16 convolutional neural network (CNN) is used for frame classification as text containing or non-text containing, has accuracy equals to 98%. The introduced system weighted average caption extraction accuracy equals to 96.15%. The correctly detected characters (CDC) average recognition accuracy using the Abbyy SDK OCR engine equals 97.75%.

Keywords Graphical text · Films application · Multimedia · Text detection · Text extraction · Subtitles frame

1 Introduction

Computer vision plays an important role to facilitate our daily life problems. Machine learning has become one of the main players in computer vision that is used to replace human beings and solve different problems like autonomous driving, object detection, object recognition, banking transactions, having smart cities and property intrusion using image processing, artificial intelligence, pattern recognition, Internet of Things (IoT), blockchain, physics, mathematics, and signals processing. Digitalization is an essential step to achieving computer vision. Analog inputs shall be converted into digital to be able to process it using the computing units. For instance, the book has to be scanned using a scanner to perform OCR. Also, a video shall be used in a digital format like Moving Picture Experts Group (MPEG) for processing in various applications of image recognition and speech recognition. There are many applications where we can apply computer vision for instance bank trading, autonomous driving, factory robots, and OCR as shown in Fig. 1. We will focus on our work on extracting and enhancing the subtitle text in film multimedia application which helps in better and faster film analysis.

2 Motivation

We will focus on multimedia applications and specifically the foreign film subtitles which can help others to build an idea about the film main story, goal, and content. This extracted text when converted to a plain text can be used to classify the film automatically as tragedy, comedy, science fiction, action, etc. We can also build an idea about how a film is influencing the audience knowing that it is social, psychological, etc. This can be performed using pattern-based networks (like DeConvolutional Neural Networks) or feature-based networks (like generative adversarial networks). Our contribution is first text extraction with high significant performance. Second, we made a generalization and tested it with four different languages. Third, the plain text from OCR is placed in a text file for further processing by film critics and the audience.

✉ Hossam Elshahaby
hossam.elshahaby@gmail.com

Mohsen Rashwan
mrashwan@cu.edu.eg

[1] Cairo University, Giza, Egypt

(a)
(b)
(c)
(d)
PLEASE TAKE NOTHING BUT PICTURES LEAVE NOTHING BUT FOOT PRINTS

Fig. 1 Computer vision applications **a** bank trading, **b** autonomous driving, **c** factory robots, **d** OCR

3 Related published work

Separation of text from its background complex graphical image is also one of the challenging topics for researchers. It contains several sub-problems including feature extraction, feature matching, text extraction, and character recognition. In some cases, like film application, the background is moving which triggers the researchers to search for algorithms that can help to resolve this problem. In this section, we will find several papers that addressed this topic in the past.

Lu and Wang (2019) position the multimedia video text for subtitle frames automatically. They improve the detection precision and the efficiency of multimedia video information. Their method uses the feature of independent video captioning (ICA) which covers the high order correlation of multimedia video data. An adaptive iterative localization algorithm is used for text localization. Detection results using this method are row recall equals 90.1%, row precision equals 95.2%, block recall equals 89.5%, and block precision equals 93% while detection results based on multi-modal fusion are row recall equals 76.9%, row precision equals 86.9%, block recall equals 74.5%, and block precision equals 84.9%.

Haq et al. (2019) use object detection to perform movie scene segmentation using a convolutional neural network (CNN). The first step segments the input movie into shots, the second step detects objects in the segmented shots, and the third step performs object-based shots matches for detecting the scene boundaries. They gather texture and shape features for shots segmentation. They apply set theory with a sliding window approach to integrate the same shots in order to decide scene boundaries. This is useful for movie trailer generation. The system showed a correct detection of 86.95%.

Yang et al. (2018) detect Chinese text captions from web videos using fully convolutional neural networks (FCN). FCN makes predictions for text line existence without a segmentation algorithm for word or character. The predictions are input to a character classifier. They fixed the drawbacks of the proposed method and remove intermediate steps using a pixel-wise classification approach. For text detection, they first sort all the word and character boxes by their y coordinates and calculate if there is a gap distance between two adjacent boxes. Second, they make a threshold to separate the text line boxes into groups according to the gap distances. Third, they group all word box groups using a boundary box. They achieved 21 frames per second on GTX1080. The proposed system does not cover multi-oriented text detection. They got on dataset 1 a precision equals 86.13%, recall equals 92.85%, and F1-score equals 89.36%. Using dataset 2, they got a precision equals to 87.31%, recall equals 92.62%, and F1-score equals 89.88%. Using dataset 3, they got a precision equals to 85.11%, recall equals 91.7%, and F1-score equals 88.28%.

Zhang et al. (2016) proposed a method for scene text detection. First, they generate the text prediction maps and geometric approaches for inclined proposals using fully convolutional network (FCN). Second, they combine the salient map and character components to estimate the text line hypotheses. Last, they predict the centroid of each character using another FCN classifier to remove the false hypotheses. The method handles text in multiple languages and fonts. The proposed method consistently achieves the performance on three text detection benchmarks. The model detects a multi-oriented scene text.

Hoang and Tabbone (2010) focuses on text extraction from graphical images. Text extraction is done from a complex background. They made the advancement of spare representation with two chosen discriminative complete dictionaries based on Undecimated Wavelet Transform and Curvelet Transform. They used the morphological component analysis (MCA). The method had better text extraction from graphics. It does not depend on text style, size, or orientation. Text extraction accuracy is 93.75%

Audithan and Chandrasekaran (2009) had a quick method for text extraction using Haar Discrete Wavelet Transform for edge detection using the Canny detector. A Canny detector extracts stroke information. Non-edges can be removed using a thresholding technique. Text edges can be connected using a morphological operator. A line feature vector was created based on an edge map and filtered. Text extraction accuracy is 94.76%.

Grover et al. (2009) detect colored text from complex background. The system is independent of contrast. Using the Sobel edge detector, they perform convolution with its mask and the image after converting the RGB image to grayscale. They eliminated non-maxima and made thresholding for weak edges. The edge image is divided into non-overlapping blocks depending on image resolution. Block classification using the defined threshold then classifies the text from the non-text. Text extraction accuracy is 94.80%.

Jung and Kim (2004) made a hybrid learning mechanism with an artificial neural network-based approach (ANNs) and a non-negative matrix factorization-based filtering (NNMF) approach to extract text from complex images. Multilayer perceptron neural network classifier increased a recall rate and a precision rate using NMF-filtering-based analysis to get connected components (CC). Text detection was performed using neural networks without having a feature extraction stage. MLPs have automatically generated a texture classifier that discriminates text regions from non-text regions based on three color bands. They learn a precise boundary between text and non-text classes using the bootstrap method. They used CC-based filtering with the NMF technique in order to overcome the locality property of the texture-based method. Processing time reduction was done using CAM shift for the video images and X–Y recursive cut algorithm for the document images. Text extraction accuracy is 91.88%.

4 Problem definition

There are subtitle texts in Films which is important when correctly recognized for various applications. The main challenge as shown in Fig. 2 is to extract text, enhance it as in Sect. 4.4, increase its resolution for the OCR engine to easily segment, and recognize the characters.

In foreign films application, there is a challenge to detect and extract the text from its graphical complex background as shown in Fig. 3. However, the resolution quality of the text is high. Many powerful OCRs are existing for the researches like Abbyy Software Development Kit (SDK), Tesseract, etc. They can help as standalone tools that support most languages, common themes, fonts, and sizes.

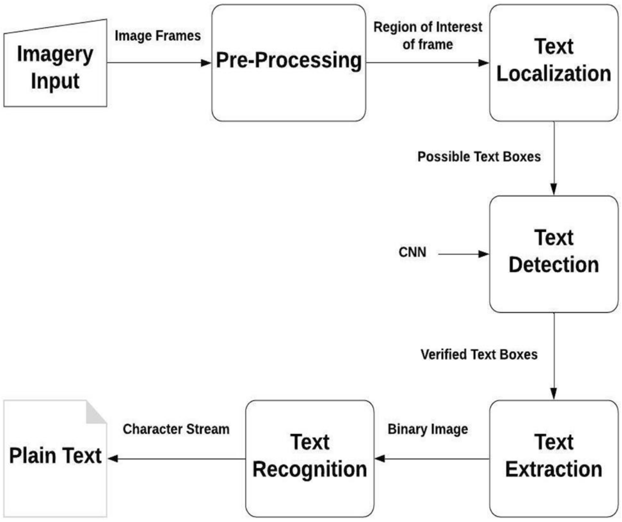
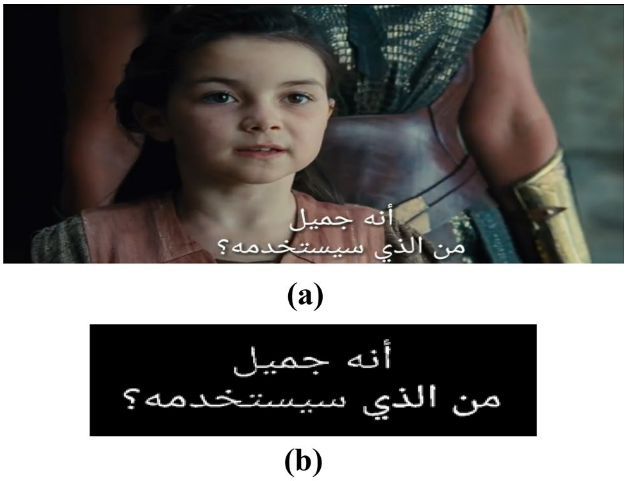


Fig. 2 Stepwise methodology for text detection/recognition



(a)

(b)

Fig. 3 Imagery text **a** graphical text input frame from movie. **b** Output result for a movie frame

5 What is imagery text?

To be able to detect text correctly, we need to define it. Imagery text is a group of edges, strokes, connected components (CCs), and texture found in image format (Ye and Doermann 2015). Normally, text related to a certain language is composed of several characters which have well-known patterns (group of features). Imagery text has several types as shown in Fig. 4. It can be caption text (overlay text or cut line text), graphical text, point-and-shoot text taken by the camera, and incidental scene text.

6 Text extraction methodologies

In order to extract text from the graphical background, we need first to process an array of an unsigned integers data frame of dimensions (h, w, 3) where h is the height and w is the width, locate the text, verify that location, and extract the text.

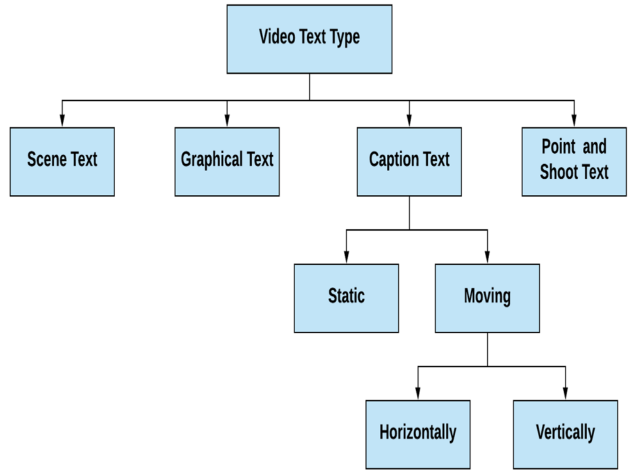


Fig. 4 Video text types

6.1 Text image localization

The video frames can be read one by one. Assuming that the subtitled text is usually placed at the lower third part of the frame, we filter the frame processing to this part to increase the performance and accuracy while decreasing the system's overall processed data. Cropping the frame will be done automatically without user intervention when this option is selected from our graphical user interface (GUI). There are some other options to select the middle, top, and full-frame just in case, the text is existing there.

6.2 Text image verification

Features extraction mainly consists of two stages which are feature selection then features dimensionality reduction. In literature features selection can be done using state vector machine (SVM), decision tree (DT), autoencoder, etc., while features dimensionality reduction can be done using principal component analysis (PCA), isomap, multilinear subspace learning, etc. We employed two different methods on our application that could show better results than others and compared them. The first one is using a geometrical method which best suites our problem nature which contains the text. It can detect edges, connected components, strokes, maximally stable extreme regions (MSER), and texture. The geometric properties that can be processed are bounding boxes, aspect ratio, eccentricity, solidity, extent, and Euler number. The advantage of using this method is that it is simple but it depends on the geometry of the text like texture, font, and size so it does not provide language-independent solution. The second one is deep learning neural network method which classifies a frame as has text or not. This method is much more powerful than the geometrical method. It has better processing time per frame and accuracy percentage as in Tables 1, 2, 3, 4, 5 and 6. It provides language independent solution for text image features extraction. However, it requires a pre-trained network which can be done easily offline. In this pre-training stage, the DNNs consist of three layers: the input layer, the hidden layer(s), and the output layer. The input layer x is mapped to the output y using

$$y = s(Wx + b), \tag{1}$$

where 'W' is the weight matrix, 'b' is the bias value, and 's' is the transfer function of Softmax which is defined by:

$$\sigma(y)_i = \frac{e^{y_i}}{\sum_{j=1}^{k} e^{y_i}}, \tag{2}$$

where i = 1, …, k and y = (y_1, …, y_k)∈R^k.

The standard exponential is applied to each y_i element of the input vector 'y', then the values are normalized by dividing by the sum of these exponentials so that the sum of the components of the output vector σ(y) is equal to 1. The cross-entropy loss function $E\ (\theta_\ell)$ can be defined as:

$$E(\theta_\ell) = -(z\log(p) + (1 - z)\log(1 - p)), \tag{3}$$

where 'ℓ' is the iteration number, 'θ' is the parameter vector, 'z' is a binary indicator (0 or 1) if class label c is the correct classification of the observation o then z equals 1, and 'p' is the predicted probability observation o of class c.

Table 1 Text or non-text classification experiments comparison using various neural nets

Network	Accuracy of test data	Average precision	Average recall	Average f-score	Training time (h)	Classification time/frame (ms)
HMLe8Net	0.9688	0.9695	0.9668	0.9681	0.2	1.65
HMAlex12 Net	0.9685	0.9665	0.9689	0.9676	1.8	6.31
HMZF13 Net	0.9794	0.9769	0.9808	0.9787	1.1	3.75
HMVGG16Net	**0.9804**	**0.9773**	**0.9827**	**0.9797**	**17.6**	**20.41**
HMGoogLe24 Net	0.9695	0.9705	0.9673	0.9688	2.8	9.21
HMRes21Net	0.9758	0.9727	0.9776	0.9750	0.5	8.97

it is bold to highlight that this designed neural network has the highest performance regarding accuracy, precision, recall, etc

Both methods can be combined together in a hybrid way to provide a robust solution for the problem.

6.3 Text image extraction

Features like a pattern or different structure can be found in an image, such as a point, line, edge, and patch. Useful features have repeatable detection, distinctive, and localizable. Using the geometrical method:

1. Compute the stroke width (Ye and Doermann 2015) metric.
2. Threshold the stroke width variation metric.
3. Process/remove regions based on this metric.
4. Determine the good candidate boundary boxes.
5. Compute their overlap ratio.
6. Find the connected text and merge them.

By performing this method on Avengers infinity war video, we got accuracy equals to 54.55% as in Table 6. While using new various deep learning networks for features extracting, to identify if a frame has text or not, like HMLe8, HMAlex12, HMZF13, HMVGG16, HMGoogLe24, and HMRes21, we got better results than the geometrical method. We performed a statistical significance test for all models as illustrated in Table 1 for the investigated samples. We chose HMVGG16 from all DNNs as it has the best results concerning the classification accuracy, recall, precision, and f-score but it consumes more pre-training time which is done offline and more processing time than other models. The HMVGG16 network model is illustrated in Fig. 5 where all its convolutional layers have a filter size of 3*3, a stride of 1, and padding of 1. It contains Convolutional_1 and Convolutional_2 layers which have a depth of "128", Max. Pooling_3 has a filter size equals 2*2 and stride of 2, Convolutional_4, Convolutional_5, and Convolutional_6 layers have a depth of "256", Max. Pooling_7 has a filter size equals 2*2 and stride of 2, Convolutional_8, Convolutional_9, and Convolutional_10 layers have a depth of "512", Max. Pooling_11 has a filter size equals 2*2 and stride of 2, fully connected_12 layer has 1024 neurons, fully connected_13 layer has 1024 neurons, fully connected_14 layer has 2 neurons, Softmax_15 layer, and classification_16 layer as in Fig. 5. We used stochastic gradient descent with momentum (SGDM) optimizer with an initial learning rate (α) of 0.0001 and a mini-batch size of 64. HMVGG16 network model consists of 16 layers. An SGDM with momentum equals 0.9 is used to update the network parameters (weights and biases) and minimize the loss function by taking small steps at each iteration in the direction of the negative gradient of the loss i.e. the algorithm can oscillate along the path of the steepest descent towards the optimum and using the momentum the oscillation is reduced as in Eq. (4):

$$\theta_{\ell+1} = \theta_{\ell} - \alpha \nabla E(\theta_{\ell}) + \gamma(\theta_{\ell} - \theta_{\ell-1}), \tag{4}$$

where α > 0 is the learning rate and γ determines the contribution of a previous gradient step to its current step.

Using the work flow chart in Fig. 6 the text extraction of text captions can be performed using its first, middle, and last appearance. The first appearance frame is resulted from subtracting this first frame with text from the one proceeding it without text. The last appearance frame is the one coming from subtracting the last coming text frame from the one just

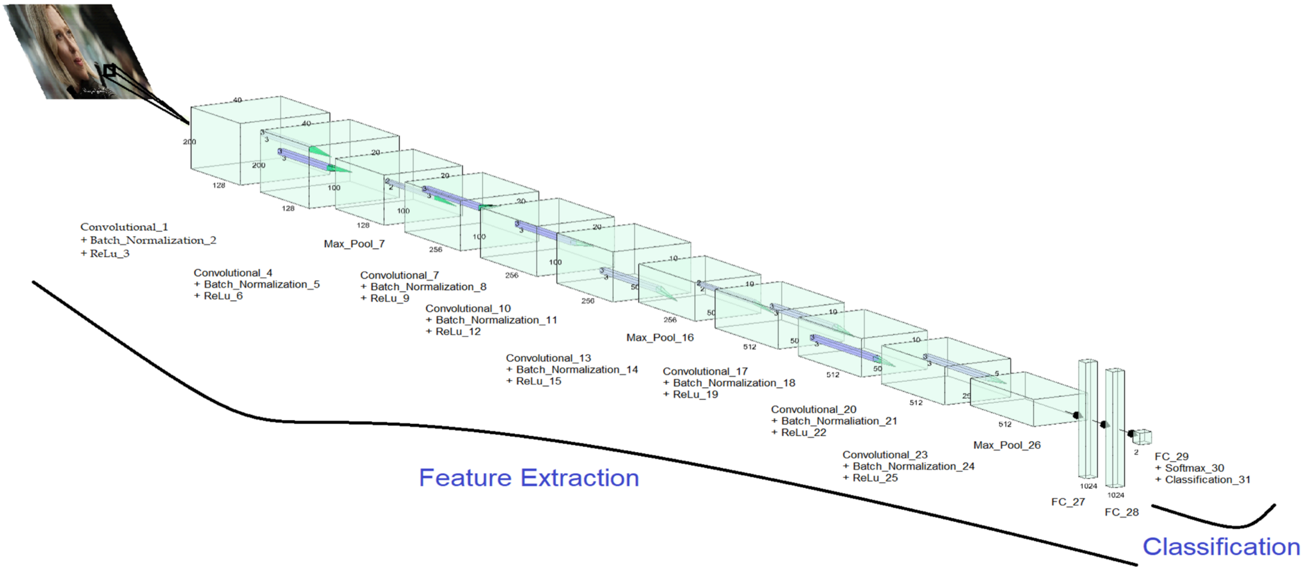


Fig. 5 HMVGG16 neural net for text or non-text classification

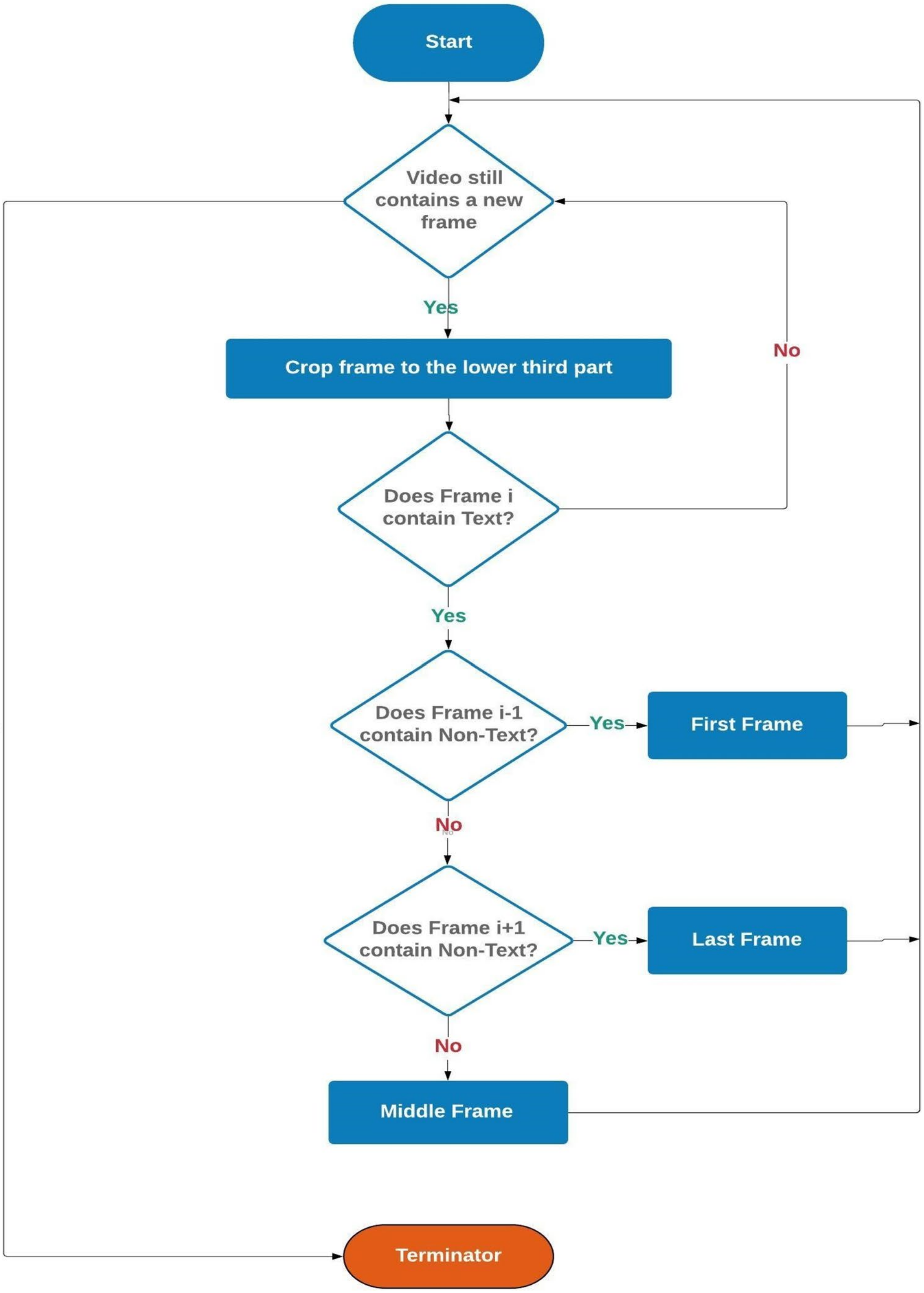

Start
Video still contains a new frame
Yes
Crop frame to the lower third part
No
Does Frame i contain Text?
Yes
Does Frame i-1 contain Non-Text?
Yes
First Frame
No
Does Frame i+1 contain Non-Text?
Yes
Last Frame
No
Middle Frame
Terminator

Fig. 6 Films proposed system flow chart

after it. The middle frame is extracted from multiplying the successive binary middle frames to get a resultant middle frame. For the binary frame data, ‘0’ represents black pixels while ‘1’ represents white pixels. The algorithm process the data coming from these three frames to get the best resultant frame which is clear and clean text extracted from the complex movie graphical background. Tracking points in films which represents the static subtitled text is done using the Kanade–Lucas–Tomasi (KLT) algorithm (Ye and Doermann 2015) as a feature-tracking algorithm. It works particularly well for tracking objects that do not change shape and for those that exhibit visual texture that is why it satisfies our need here. Initial points for tracking the graphical subtitled text caption are obtained using the minimum eigen features (MEF) algorithm (Shi and Tomasi 1994).

Using the forward–backward error threshold as an automatic algorithm for the detection of tracking failures, the tracker tracks each point from the previous to the current frame. Then, it tracks the same points back to the previous frame as seen below in Fig. 7. The object calculates the bidirectional error. This value is the distance in pixels from the original location of the points to the final location

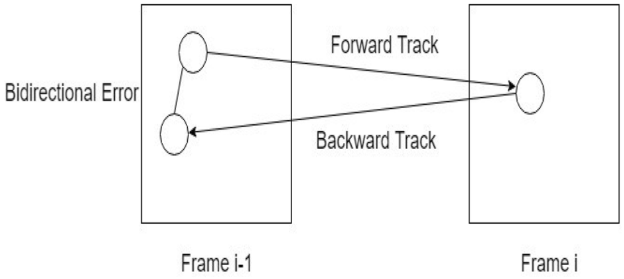


Fig. 7 Bidirectional error method

after the backward tracking. The corresponding points are considered invalid when the error is greater than the value set for this property (Ye and Doermann 2015). This method extends the range of the estimate of disparity (h) by smoothing the images. In order to combine the various values of "h". We simply average them:

$$h \approx \sum_x \frac{G(x) - F(x)}{F(x)} / \sum_x 1 \tag{5}$$

$$h_{k+1} = h_k + \sum_x \frac{w(x)\left[G(x) - F(x + h_k)\right]}{F(x + h_k)} / \sum_x w(x). \tag{6}$$

The estimates sequence of h (Ye and Doermann 2015) will converge to the best h.

Where:

F (x): Pixel values at each x location function in the image. G (x): Pixel values at each x location function in the next image.

W (x): Weighting function.

F (x + h_k): Pixel values at each (x + h) location function in image.

Using the bidirectional error through successive frames is an effective method to eliminate points that could not be tracked in a reliable way. We can only keep points that contain the highest value for the system. However, the bidirectional error requires more additional computation to be done.

By applying this KLT algorithm to track static pixels through the successive frames with calculating the bidirectional error using the predefined forward–backward error threshold, we can get an output result as shown in Fig. 8. It is obvious that the algorithm successfully tracked the static pixels representing the text. It accurately creates points showing the location of the subtitled text. The algorithm has a stabilization effect for the points while tracking them through different frames where they exist. This performs text stabilization which accordingly enhances the quality of extracted text from the films.

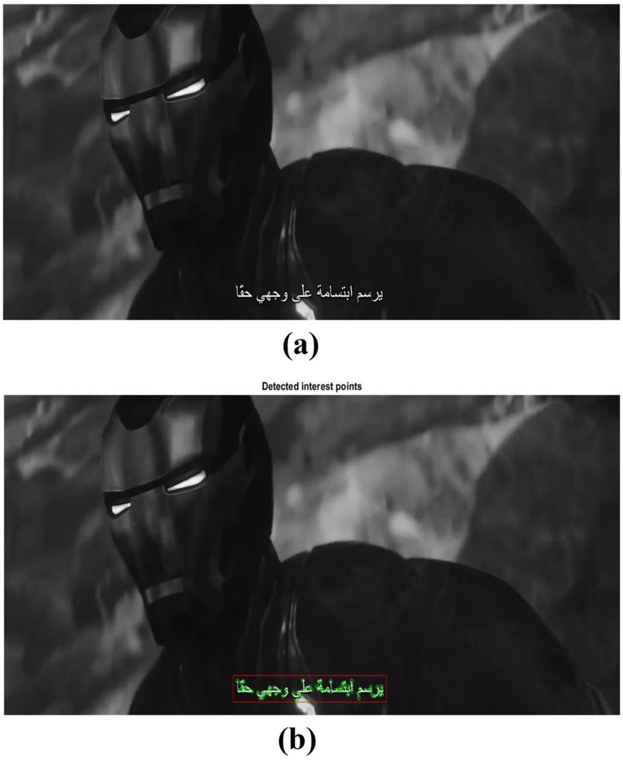


Fig. 8 Imagery text **a** original text input frame from movie. **b** Output result for pixels tracked for successive movie frames

6.4 Text image enhancement

Image enhancement techniques are applied before text image localization by adjusting contrast, color intensity, noise filtering, illumination equalization, etc. It is also applied after text image extraction by using multi-frame integration algorithm and increasing image resolution from 70 dots per inch (dpi) to 300 dpi.

7 Error handling and correction

It includes handling the neural network classification errors which may misclassify a text frame as a non-text frame or vice versa in addition to the human translation errors done by the unprofessional translators related to adding successive different text without breaks of frames in between without text. One of the populist techniques to be used is the correlation-based technique. It could be used to decide whether it is truly a non-text frame or it is an error to resolve classification errors. It can also be used to decide whether it is truly the same text frame or a totally new one to resolve translation issues as shown in Fig. 9.

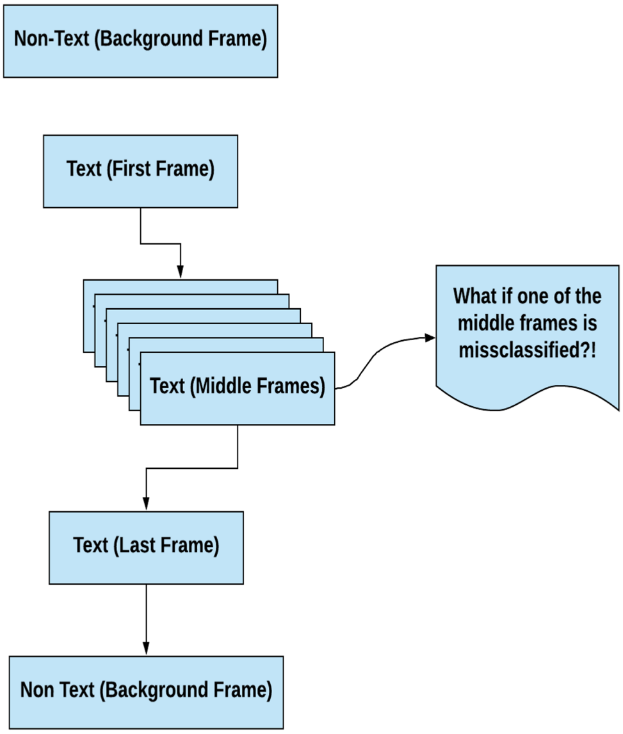

Non-Text (Background Frame)
Text (First Frame)
Text (Middle Frames)
What if one of the middle frames is missclassified?!
Text (Last Frame)
Non Text (Background Frame)

Fig. 9 Classification error corner case handling

8 Dataset

A newly self-created dataset called “FiViD” in http://tc11.cvc.uab.es/datasets/FiViD_1 is collected with different fonts, styles, sizes, etc. to evaluate the full system as in Table 4. and for training and testing purposes as in Tables 2 and 3. To access all our research artifacts, please visit https://github.com/helshahaby/Television-Movies-Artifacts.

9 Evaluation criteria

We use Matlab® for the training of the deep neural nets and its evaluation after tuning its hyperparameters like learning rate, number of epochs, optimizer used, minimum batch size, etc. as in. Fig. 10. Our environment has CUDA® enabled NVIDIA® GPU GeForce GTX 960M 4 GB, Intel® CPU Core i7-6700HQ with 2.6 GHz, and 16 GB RAM.

9.1 Execution performance

As in Table 5, we compared the geometrical method and two neural network methods which are the HMLe8 network and HMVGG16 network using Avengers infinity war Video to check the processing time of them. HMLe8 net proved to have the lowest execution time as it is the simplest neural network.

Table 2 Testing dataset for text classification

Film	# of frames	Dimensions width × height
24 h to live	3254	1280 × 720
Avengers infinity war	3361	1280 × 532
Here after	3250	480 × 220
Inferno off teaser	1382	1280 × 720
Interstellar	2610	1280 × 720
Total frames		**13,857**

Table 3 Training dataset for text classification

Film	# of frames	Dimensions width × height
Wonder woman	4104	1280 × 720
Black Panther	2449	1280 × 530
Passengers	4238	1280 × 720
Assassin creed	2909	1280 × 720
Atomic blonde	4177	1280 × 720
Divergent	1706	1280 × 534
Mother	2909	640 × 350
Spider man	3787	1280 × 720
Stronger	3412	640 × 350
Suicide squad	3932	1280 × 720
Dunkirk	3922	1280 × 720
Total frames		**37,545**

Table 4 System evaluation dataset for text extraction

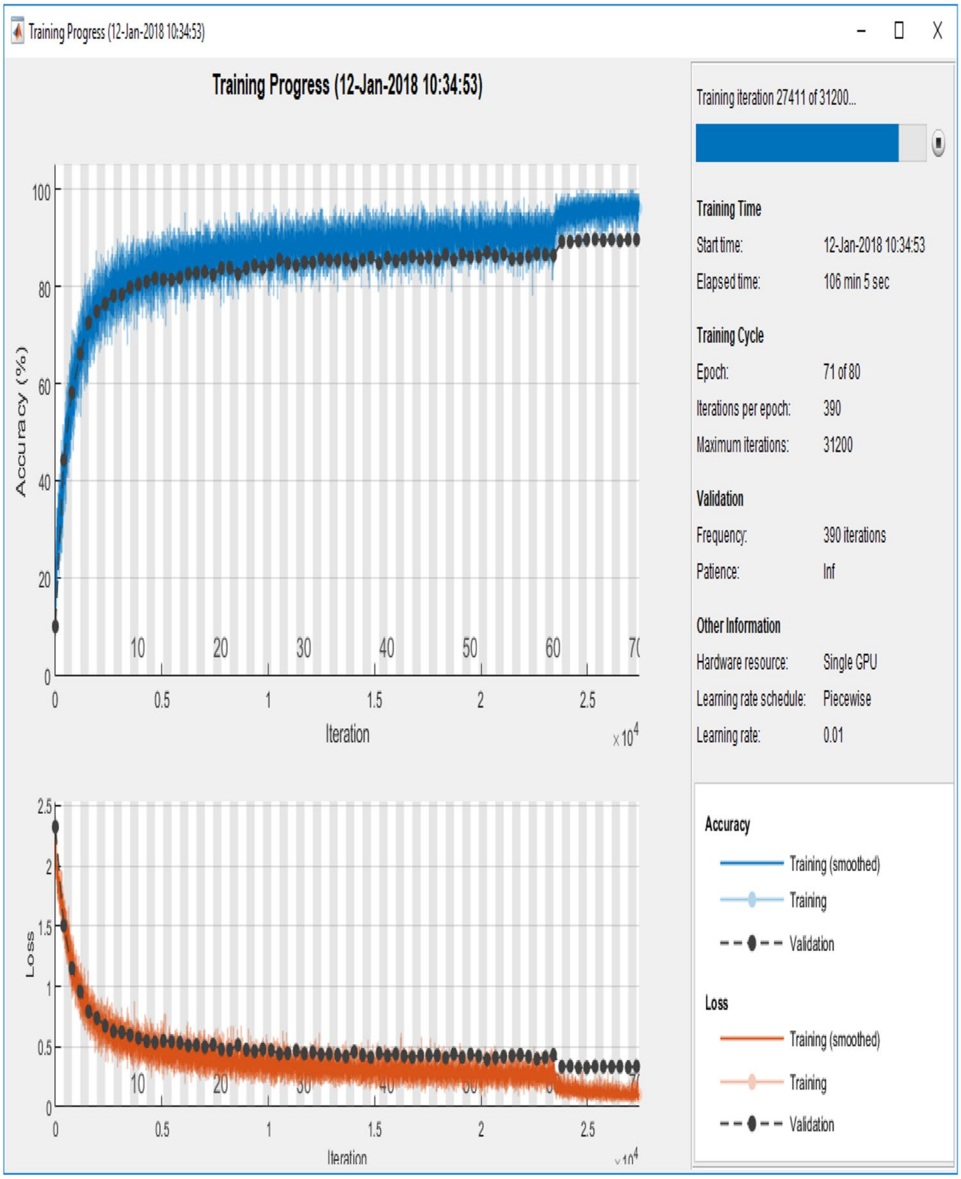
Film	# of frames	Language
Assassin creed 2016	2726	Arabic
MorrisAusAmerika	2835	Arabic
Kong Skull Island	3286	Arabic
Avengers infinity war	3361	Arabic
Venom	5824	Arabic
Hebrew film	3581	Hebrew
Legend of Naga Pearls	2854	Arabic Chinese
Sabrina	2805	English
Total frames		**27,272**

9.2 Text extraction performance

Using the same video as in Sect. 9.1, we compared the geometrical method and the same two deep learning methods to know their text extraction accuracy as in Table 6. HMVGG16 net has the highest text extraction accuracy.

The following parameters can be used to measure the system’s performance. These parameters are N_G, N_C, N_I, and N_R where:

Fig. 10 Deep learning neural net training in Matlab tool



Training iteration 27411 of 31200...

Training Time
Start time: 12-Jan-2018 10:34:53
Elapsed time: 106 min 5 sec

Training Cycle
Epoch: 71 of 80
Iterations per epoch: 390
Maximum iterations: 31200

Validation
Frequency: 390 iterations
Patience: Inf

Other Information
Hardware resource: Single GPU
Learning rate schedule: Piecewise
Learning rate: 0.01

Table 5 Processing time/frame comparison

Geometrical	HMLe8Net (ms)	HMVGG16 (ms)
366 ms	59.13	115.36

Table 6 Percentage of text extraction accuracy comparison

Geometrical	HMLe8Net (%)	HMVGG16 (%)
54.55%	85	90.9

N_G: Number of ground truth graphical captions in the film.

N_C: Number of graphical captions correctly detected.

N_I: Number of graphical captions wrongly inserted.

Table 7 System performance using evaluation dataset

Film	N_G	N_C	N_I	N_R
Assassin Creed 2016	10	10	2	0
Kong Skull Island	19	19	0	0
Avengers infinity war	27	27	1	0
Hebrew	12	12	1	0
Legend of Naga Pearls	13	11	0	2
Sabrina	19	19	0	0
Total	**100**	**98**	**4**	**2**

N_R: Number of graphical captions wrongly missed.

For the reason that the films' captions contain four different languages, we can perform the weighted mean using Eq. (7) for all the system metrics indicated above (Table 7):

$$Weightedmean = \frac{\sum_k (x_k * w_k)}{\sum_k w_k} \tag{7}$$

The system metrics equations are calculated as:

$$Accuracy = \sum_k \frac{N_C}{N_G} = 96.15\% \tag{8}$$

$$Precision = \frac{\sum_k N_C}{\sum_k N_C + \sum_k N_I} = 93.97\% \tag{9}$$

$$Recall = \frac{\sum_k N_C}{\sum_k N_C + \sum_k N_R} = 94.79\% \tag{10}$$

$$f - Score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = 94.38\% \, . \tag{11}$$

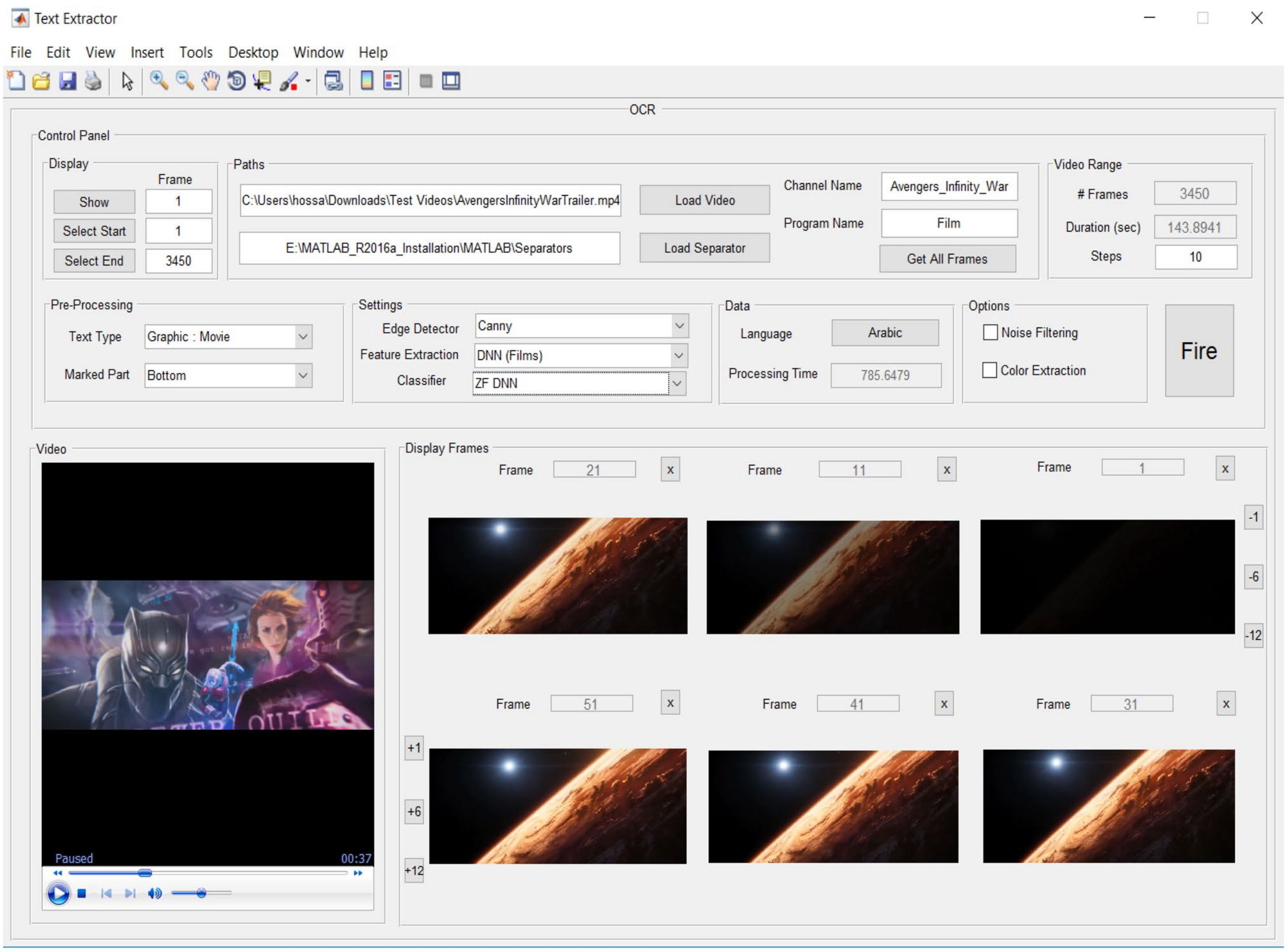
A new TextExtractor GUI is created to facilitate the analysis and evaluation of the proposed system as shown in Fig. 11. It shows all important video information once it is loaded by the user. The extracted text images are saved in image format and also gathered placed in a word format file to be easily accessed in one document.

9.3 Text recognition performance

Using Abbyy Cloud SDK as a COTS software tool with our dataset, we recognize characters from four different languages like Arabic, Latin, Hebrew, and Chinese. The overall average accuracy is equal to 97.75%.

10 Discussion

We propose two techniques to solve the problem of extracting subtitles from movie videos. The first one is geometrical based on style, font size, and language type. While the



Display: Frame — Show 1; Select Start 1; Select End 3450

Paths: C:\Users\hossa\Downloads\Test Videos\AvengersInfinityWarTrailer.mp4

E:\MATLAB_R2016a_Installation\MATLAB\Separators

Channel Name: Avengers_Infinity_War

Program Name: Film

Video Range: # Frames 3450; Duration (sec) 143.8941; Steps 10

Pre-Processing: Text Type Graphic : Movie; Marked Part Bottom

Settings: Edge Detector Canny; Feature Extraction DNN (Films); Classifier ZF DNN

Data: Language Arabic; Processing Time 785.6479

Options: □ Noise Filtering □ Color Extraction

Paused 00:37

Frame 21; Frame 11; Frame 1

Frame 51; Frame 41; Frame 31

Fig. 11 Text extractor GUI new tool for better usability

second one based on a deep neural network (DNN) that classifies the frame inside the video as containing text or not. We proposed several deep neural networks like HMLe8, HMAlex12, HMZF13, HMVGG16, HMGoogLe24, and HMRes21. Using statistical analysis, we found that the most powerful network is the HMVGG16 model that is why we chose it from all other models. HMVGG16 better solves our problem in terms of accuracy, recall, precision, and f-score but it has longer pre-training time processing time while HMLe8 has lower accuracy but better timing considerations. For HMVGG16, the learning rate (α) is adjusted to be 0.0001, the minimum batch size is 64 which is high to have larger gradient steps and larger variance in distance. The system has the advantage of working with multiple languages and it is evaluated using four different languages. In addition, some of the evaluated films can have two different language translations at the same time. One challenging problem occurs when translating text is appearing gradually and is removed in a fading way which makes it difficult for this algorithm to subtract the frame with text with the same frame containing the background without the text. This limitation when it occurs decreases the system accuracy and makes it difficult for the OCR to recognize faded unclear text. This may cause a system failure in some cases. The system also relies on a pre-trained model (HMVGG16) which may underperform with untrained text styles and fonts. However, our proposed method works particularly well in extracting text from a complex background with an accuracy of 96.15% that shows better performance than the other proposed methods in the open literature.

The computational complexity of the proposed system using asymptotic measurements is $O\ (N+pn_{l1}+n_{l1}n_{l2}+\ldots n_{l15}n_{l16})$ where ‘p’ is the number of features and ‘n_{li}’ is the number of neurons in layer ‘i’ in the neural network. Using our evaluation dataset in Table 4 above where the film captions contain four different languages, we can perform the weighted mean for all the system metrics. We can find weighted average insertion error for additionally added captions equal to 6.68%, we can compute the weighted average deletion error for missed captions equals to 5.36%. We can calculate the average weighted accuracy equals to 96.15%, average weighted precision equals to 93.97%, average weighted recall equals to 94.79%, and the average weighted f-score equals to 94.38%. We finally use Abbyy SDK as a black box OCR engine to qualify our system from its intended outcome point of view which is to have a clear clean plain text for the users of the system. The overall average accuracy is equal to 97.75%.

Lu and Wang (2019) improved automatic positioning accuracy using the ICA feature which is a stoke segment that constitutes the image caption base. The adaptive iterative localization algorithm has strong adaptability to complex changes of video frames, faster, and more accurate detection and positioning of multimedia video lines and blocks. However, the method has a lack of completeness of feature extraction and the accuracy of selection and positioning of the candidate regions needs to be improved.

Haq et al. (2019) segment scene boundaries in movies using a convolution neural network. However, the method can be used for keyframes extraction for indexing and retrieval, video abstraction, skims selection, and trailer generation.

Yang et al. (2018) proposed a deep CNN model that is optimized by SGD. They solved the problem of error accumulation during denoise and binarization. However, the method is sensitive to text style, size, etc.

Zhang et al. (2016) detects scene text with multi-orientation using FCN. The method handles different text orientations, languages, and styles. Failure occurs with extremely low contrast, curvature, strongly reflect light, too close text lines, or a tremendous gap between characters. It cannot be used in a real-time environment.

Hoang and Tabbone (2010) employ the MCA method using undecimated wavelet and curvelet transforms and promote spare representation. The system advantage is that it is invariant to different font styles, sizes, and orientations. Text extraction accuracy is 93.75%

Audithan and Chandrasekaran (2009) used Haar Discrete Wavelet Transform (DWT) for edge detection using the Canny detector. Text. Haar is the fastest among all other wavelets because its coefficients are either 1 or − 1. The method suppresses false alarms. However, it has a limitation when the gradient color of text and background are quite close. Text extraction accuracy is 94.76%.

Grover et al. (2009) results were also well with high sensitivity and low false alarm rate. It has a limitation when the gradient of intensities of text and image are quite similar. They used a Sobel Edge Detector and got text extraction accuracy of 94.80%.

Jung and Kim (2004) combined the neural net-based detection with NMF based filtering. The main drawback is in its locality property i.e. it does not consider the text outside the window. However, they adopt CAMShift to enhance time performance. Text extraction accuracy is 91.88%.

11 Conclusion and future work

In this research, the common problem of imagery text detection and enhancement from videos is discussed. Proposed solutions for processing text videos to detect text automatically and extract it from images are implemented. Different machine learning techniques like HMVGG16 and HMLe8 networks are applied to identify graphical text in films application using deep convolutional neural networks. The point tracking technique is adopted for text extraction from its complex background. A new self-created dataset "FiViD" is created to be used as a benchmark. The HMVGG16 deep CNN network which is used for frame classification as text containing or non-text containing has accuracy equals 98%. Using "film videos dataset" to evaluate the graphical caption extraction, the weighted average caption extraction accuracy is equal to 96.15%, insertion error equals to 6.68%, deletion error is equal to 5.36%, precision is equal to 93.97%, recall is equal to 95.27%, and CDC recognition average accuracy is equal to 97.75%. The future work in our film multimedia application is to make our own OCR and decrease execution timing for the frame classifier to run in a real-time environment. Also, we can train our text classifier model to support more languages like Russian, Indian, etc. We can enable the user to translate the existing subtitle plain text to any other selected language of user choice.

Acknowledgements I would like to thank God for his help. Special thanks to the RDI team, Dr. Sven Dickinson, and my faculty department members for supporting me with their experience and data set used in my research.

References

Alves W, Hashimoto R (2010) Text regions extracted from scene images by ultimate attribute opening and decision tree classification. In: Proceedings of the 23rd Sibgrapi conference on graphics, patterns, and images

Audithan S, Chandrasekaran RM (2009) Document text extraction from document images using Haar discrete wavelet transform. Eur J Sci Res 36(04):502–512

Cho H, Sung M, Jun B (2016) Canny text detector: fast and robust scene text localization algorithm. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3566–3573

Dai J, Li Y, He K, Sun J (2016) R-FCN: object detection via region-based fully convolutional networks. In: advances in neural information processing systems, pp 379–387

Gidaris S, Komodakis N (2015) Object detection via a multi-region and semantic segmentation-aware CNN model. In: Proceedings of the IEEE international conference on computer vision, pp 1134–1142

Gomez L, Karatzas D (2017) Text proposals: a text specific selective search algorithm for word spotting in the wild. Pattern Recogn 70:60–74

Gorinski P, Lapata M (2018) What's this movie about? A joint neural network architecture for movie content analysis. In: University of Edinburgh, Proceedings of NAACL-HLT, pp 1770–1781

Grover S, Arora K, Mitra S (2009) Text extraction from document images using edge information. In: IEEE India Council Conference

Gupta A, Vedaldi A, Zisserman A (2016) Synthetic data for text localization in natural images. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2315–2324

Haq I, Muhammad K, Hussain T, Kwon S, Sodanil M, Baik S, Lee M (2019) Movie scene segmentation using object detection and set theory. Int J Distrib Sens Netw 15(6)

He K, Zhang X, Ren S, Sun J (2016a) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778

He T, Huang W, Qiao Y, Yao J (2016b) Text attentional convolutional neural network for scene text detection. IEEE Trans Image Process 25(6):2529–2541

He P, Huang W, He T, Zhu Q, Qiao Y, Li X (2017) Single shot text detector with regional attention. In: Computer vision and pattern recognition, Cornell University, arXiv:1709.00138

Hesham M, Hani B, Fouad N, Amer E (2018) Smart trailer: automatic generation of movie trailer using only subtitles. In: First international workshop on deep and representation learning (IWDRL), IEEE, pp 26–30

Hoang T, Tabbone S (2010) Text extraction from graphical document images using sparse representation. In: Proceedings of the 9th IAPR international workshop on document analysis systems, pp 143–150

https://pixabay.com/vectors/bitcoin-money-cryptocurrency-4851383/. Accessed 28 Sept 2020

https://www.dreamstime.com/photos-images/autonomous-car.html. Accessed 28 Sept 2020

https://www.freepik.com/premium-photo/engineer-check-control-welding-robotics-automatic-arms-machine_5284742.htm. Accessed 28 Sept 2020

https://www.robots.ox.ac.uk/~vgg/software/textspot/. Accessed 10 June 2020

Huang W, Qiao Y, Tang X (2014) Robust scene text detection with convolution neural network induced MSER trees. In: European conference on computer vision, Springer, Zurich, pp 497–511

Indermühle E, Liwicki M, Bunke H (2010) IAMonDo-database: an online handwritten document database with non-uniform contents. In: Proceedings of the 9th IAPR international workshop on document analysis systems (DAS '10), pp 97–104

Jaderberg M, Simonyan K, Vedaldi A, Zisserman A (2016) Reading text in the wild with convolutional neural networks. Int J Comput Vis 116(1):1–20

Jung K, Kim E (2004) Automatic text extraction for content-based image indexing. In: Proceedings of PAKDD, pp 497–507

Kong T, Yao A, Chen Y, Sun F (2016) Hypernet: towards accurate region proposal generation and joint object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 845–853

Liao M, Shi B, Bai X, Wang X, Liu W (2017) Textboxes: a fast text detector with a single deep neural network. In: AAAI, pp 4161–4167

Liu X, Samarabandu J (2006) Multiscale edge-based text extraction from complex images. In: Proceedings of the international conference of multimedia and Expo, pp 1721–1724

Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440

Lu Q, Wang Y (2019) Automatic text location of multimedia video for subtitle frame. J Ambient Intell Humaniz Comput

Moradi M, Mozaffari S, Orouji A (2010) Farsi/Arabic text extraction from video images by corner detection. In: 2010 6th Iranian conference on machine vision and image processing, pp 1–6

Nagabhushan P, Nirmala S (2009) Text extraction in complex color document images for enhanced readability. Intell Inf Manag 2:120–133

Neumann L, Matas J (2012) Real-time scene text localization and recognition. In: Computer vision and pattern recognition (CVPR) IEEE conference, pp 3538–3545

Noh H, Hong S, Han B (2015) Learning deconvolution network for semantic segmentation. In: Proceedings of the IEEE international conference on computer vision, Santiago: IEEE Computer Society, pp 1520–1528

Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the

IEEE conference on computer vision and pattern recognition, pp 779–788

Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp 91–99

Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252

Shelhamer E, Long J, Darrell T (2017) Fully convolutional networks for semantic segmentation. IEEE Trans Pattern Anal Mach Intell 39(4):640–651

Shi J, Tomasi C (1994) Good features to track. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 593–600

Shivakumara P, Dutta A, Pal U, Tan C (2010) A new method for handwritten scene text detection in video. In: International conference on frontiers in handwriting recognition, pp 16–18

Shrivastava A, Gupta A, Girshick R (2016) Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas: IEEE Computer Society, arXiv:1604.03540

Sun L, Huo Q, Jia W, Chen K (2015) A robust approach for text detection from natural scene images. Pattern Recogn 48(9):2906–2920

Tian S, Pan Y, Huang C, Lu S, Yu K, Tan C (2015) Text flow: a unified text detection system in natural scene images. In: Proceedings of the IEEE international conference on computer vision, pp 4651–4659

Tian Z, Huang W, He T, He P, Qiao Y (2016) Detecting text in natural image with connectionist text proposal network. In: European conference on computer vision, pp 56–72

Vijayakumar V, Nedunchezhianm R (2011) A novel method for super imposed text extraction in a sports video. Int J Comput Appl 15(1):1

Xiang D, Yan H, Chen X, Cheng Y (2010) Offline Arabic handwriting recognition system based on HMM. In: 2010 3rd International conference on computer science and information technology

Yang C, Pei W, Wu L, Yin X (2018) Chinese text-line detection from web videos with fully convolutional networks. Big Data Anal 3(2):1

Ye Q, Doermann D (2015) Text detection recognition in imagery: a survey. IEEE Trans Pattern Anal Mach Intell 37(7):1480–1500

Yin XC, Pei WY, Zhang J, Hao H (2015) Multi-orientation scene text detection with adaptive clustering. IEEE Trans Pattern Anal Mach Intell 37(9):1930–1937

Zamberletti A, Noce L, Gallo I (2014) Text localization based on fast feature pyramids and multi-resolution maximally stable extremal regions. In: Asian conference on computer vision, pp 91–105

Zhang Z, Shen W, Yao C, Bai X (2015) Symmetry based text line detection in natural scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2558–2567

Zhang Z, Zhang C, Shen W, Yao C, Liu W, Bai X (2016) Multi-oriented text detection with fully convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Las Vegas: IEEE Computer Society, pp 4159–4167

Zhang S, Liu Y, Jin L, Luo C (2018) Feature enhancement network: a refined scene text detector. In: Thirty-second AAAI conference on artificial intelligence (AAAI-18), pp 2612–2619

Zhong Z, Jin L, Zhang S, Feng Z (2016) DeepText: a unified framework for text proposal generation and text detection in natural images. In: Computer vision and pattern recognition, Cornell University, arXiv:1605.07314

Zhou X, Yao C, Wen H, Wang Y, Zhou S, He W, Liang J (2017) EAST: an efficient and accurate scene text detector. In: Computer vision and pattern recognition, Cornell University, arXiv:1704.03155

Zhu Y, Yao C, Bai X (2016) Scene text detection and recognition: recent advances and future trends. Front Comput Sci 10(1):19–36

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.