Annotation of English scientific articles (21.02.2024)

We need to find the following lines in the scientific articles:

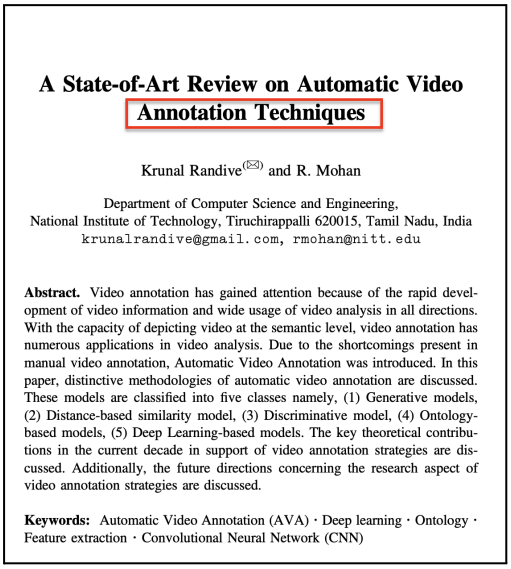
- Title of the article (**title**)
- Author (or authors) (**author**)
- Affiliation (or affiliations) (**affiliation**)
- Sections, subsections, etc. with their content (**named_item, raw_text**)
- References (**reference**)
- Captions of images, tables, listings, etc. (**caption**)

For this purpose, we solve the task of line classification.
Input: a document page with a line highlighted by a bounding box. The line should be annotated with one of the following types:

- **All** lines of the article title – **title;**
- **All** lines that contain authors names – **author;**
- **All** lines that contain affiliations and other information about the authors – **affiliation;**
- Lines with sections titles – **named_item;**
- **First** line of each reference (in the section References) – **reference**;
- Content (text) of sections or references – **raw_text;**
- All lines of image/table/… caption – **caption**;
- Other highlighted non-textual elements and page numbers should be marked as **other.**

Title

Title is located at the beginning of the article, it contains the name of the article. Each line of the article title should be labeled as **title**.
Examples:



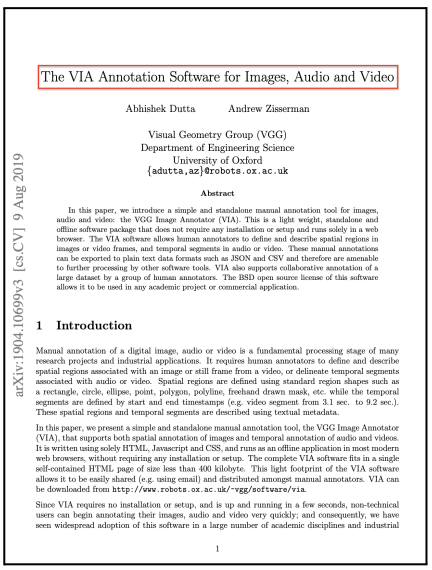
A State-of-Art Review on Automatic Video Annotation Techniques

Krunal Randive(✉) and R. Mohan

Department of Computer Science and Engineering,
National Institute of Technology, Tiruchirappalli 620015, Tamil Nadu, India
krunalrandive@gmail.com, rmohan@nitt.edu

Abstract. Video annotation has gained attention because of the rapid development of video information and wide usage of video analysis in all directions. With the capacity of depicting video at the semantic level, video annotation has numerous applications in video analysis. Due to the shortcomings present in manual video annotation, Automatic Video Annotation was introduced. In this paper, distinctive methodologies of automatic video annotation are discussed. These models are classified into five classes namely, (1) Generative models, (2) Distance-based similarity model, (3) Discriminative model, (4) Ontology-based models, (5) Deep Learning-based models. The key theoretical contributions in the current decade in support of video annotation strategies are discussed. Additionally, the future directions concerning the research aspect of video annotation strategies are discussed.

Keywords: Automatic Video Annotation (AVA) · Deep learning · Ontology · Feature extraction · Convolutional Neural Network (CNN)



The VIA Annotation Software for Images, Audio and Video

Abhishek Dutta Andrew Zisserman

Visual Geometry Group (VGG)
Department of Engineering Science
University of Oxford
{adutta,az}@robots.ox.ac.uk

Abstract

In this paper, we introduce a simple and standalone manual annotation tool for images, audio and video: the VGG Image Annotator (VIA). This is a light weight, standalone and offline software package that does not require any installation or setup and runs solely in a web browser. The VIA software allows human annotators to define and describe spatial regions in images or video frames, and temporal segments in audio or video. These manual annotations can be exported to plain text data formats such as JSON and CSV and therefore are amenable to further processing by other software tools. VIA also supports collaborative annotation of a large dataset by a group of human annotators. The BSD open source license of this software allows it to be used in any academic project or commercial application.

1 Introduction

Manual annotation of a digital image, audio or video is a fundamental processing stage of many research projects and industrial applications. It requires human annotators to define and describe spatial regions associated with an image or still frame from a video, or delineate temporal segments associated with audio or video. Spatial regions are defined using standard region shapes such as a rectangle, circle, ellipse, point, polygon, polyline, freehand drawn mask, etc. while the temporal segments are defined by start and end timestamps (e.g. video segment from 3.1 sec. to 9.2 sec.). These spatial regions and temporal segments are described using textual metadata.

In this paper, we present a simple and standalone manual annotation tool, the VGG Image Annotator (VIA), that supports both spatial annotation of images and temporal annotation of audio and videos. It is written using solely HTML, Javascript and CSS, and runs as an offline application in most modern web browsers, without requiring any installation or setup. The complete VIA software fits in a single self-contained HTML page of size less than 400 kilobyte. This light footprint of the VIA software allows it to be easily shared (e.g. using email) and distributed amongst manual annotators. VIA can be downloaded from http://www.robots.ox.ac.uk/~vgg/software/via.

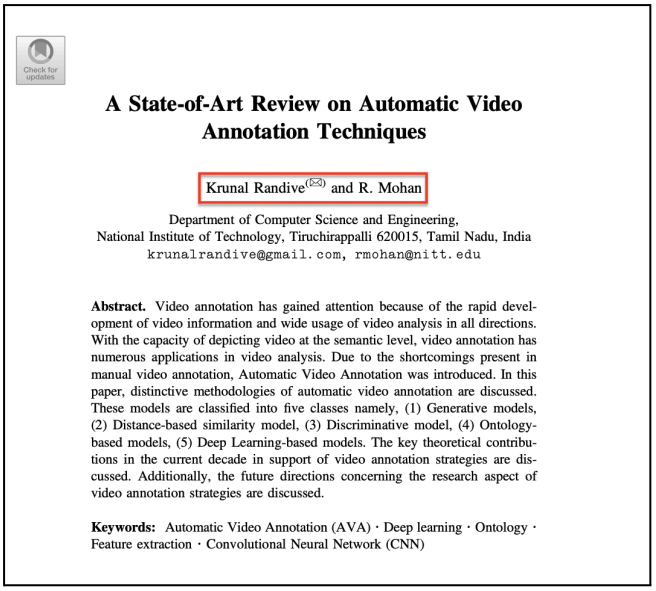
Since VIA requires no installation or setup, and is up and running in a few seconds, non-technical users can begin annotating their images, audio and video very quickly; and consequently, we have seen widespread adoption of this software in a large number of academic disciplines and industrial

1

Author

Author list is located after the title. Each line containing authors' names should be labeled as **author**.

Examples:



A State-of-Art Review on Automatic Video Annotation Techniques

Krunal Randive(✉) and R. Mohan

Department of Computer Science and Engineering,
National Institute of Technology, Tiruchirappalli 620015, Tamil Nadu, India
krunalrandive@gmail.com, rmohan@nitt.edu

Abstract. Video annotation has gained attention because of the rapid development of video information and wide usage of video analysis in all directions. With the capacity of depicting video at the semantic level, video annotation has numerous applications in video analysis. Due to the shortcomings present in manual video annotation, Automatic Video Annotation was introduced. In this paper, distinctive methodologies of automatic video annotation are discussed. These models are classified into five classes namely, (1) Generative models, (2) Distance-based similarity model, (3) Discriminative model, (4) Ontology-based models, (5) Deep Learning-based models. The key theoretical contributions in the current decade in support of video annotation strategies are discussed. Additionally, the future directions concerning the research aspect of video annotation strategies are discussed.

Keywords: Automatic Video Annotation (AVA) · Deep learning · Ontology · Feature extraction · Convolutional Neural Network (CNN)

TencentPretrain: A Scalable and Flexible Toolkit for Pre-training Models of Different Modalities

Zhe Zhao[1*], Yudong Li[2], Cheng Hou[1], Jing Zhao[1], Rong Tian[1], Weijie Liu[1], Yiren Chen[1], Ningyuan Sun[1], Haoyan Liu[1], Weiquan Mao[1], Han Guo[1], Weigang Guo[1], Taiqiang Wu[1], Tao Zhu[1], Wenhang Shi[3], Chen Chen[1], Shan Huang[1], Sihong Chen[1], Liqun Liu[1], Feifei Li[1], Xiaoshuai Chen[1], Xingwu Sun[1], Zhanhui Kang[1], Xiaoyong Du[3], Linlin Shen[2], Kimmo Yan[1]

[1] Tencent AI Lab
[2] School of Computer Science and Software Engineering, Shenzhen University
[3] School of Information and DEKE, MOE, Renmin University of China

Abstract

Recently, the success of pre-training in text domain has been fully extended to vision, audio, and cross-modal scenarios. The proposed pre-training models of different modalities are showing a rising trend of homogeneity in their model structures, which brings the opportunity to implement different pre-training models within a uniform framework. In this paper, we present TencentPretrain, a toolkit supporting pre-training models of different modalities. The core feature of TencentPretrain is the modular design. The toolkit uniformly divides pre-training models into 5 components: ***embedding***, ***encoder***, ***target embedding***, ***decoder***, and ***target***. As almost all of common modules are provided in each component, users can choose the desired modules from different components to build a complete pre-training model. The modular design enables users to efficiently reproduce existing pre-training models or build brand-new one. We test the toolkit on text, vision, and audio benchmarks and show that it can match the performance of the original implementations.

is successful in the field of text, is increasingly being applied to the vision and audio modalities. (Dosovitskiy et al., 2020; Gulati et al., 2020). Table 1 lists the commonly used pre-training models and their modules.

The trend towards homogeneity in pre-training models is becoming more apparent, which makes it possible to integrate them into a uniform framework. A representative work in this direction is Huggingface Transformers (Wolf et al., 2020), which exploits a non-modular design mode. For each pre-training model in Huggingface Transformers, several separate classes are created, and the code is not refactored with additional abstractions. Users can develop their pre-training models independently which is useful to collaborative development in the community. However, in this design mode, users need to implement the model from scratch when adding a new pre-training model, requiring considerable code work. In addition, with the increased number of pre-training models, the number of classes and lines of code also increases linearly. Codes with the same function may be written many times, which degrades the readability and maintainability of the project.

In response to shortcomings of non-modular design mode, we introduce TencentPretrain, a modular toolkit specially designed for pre-training models of different modalities. As shown in Figure 1, TencentPretrain has five components, namely ***embedding***, ***encoder***, ***target embedding***, ***decoder***, and ***target***. Among them, ***target embedding*** and ***decoder*** components are optional, since the targets of many pre-training models do not involve sequence decoding (Zhang et al., 2020; Lewis et al., 2020). TencentPretrain is hierarchical modular designed with two degrees of freedom. At component level, users are free to combine modules within a component, for example, combining multiple modules in ***target*** component to perform multi-task pre-training (Lan et al., 2019; Sun et al., 2020). At

1 Introduction

Pre-training on large-scale data and then fine-tuning on downstream tasks has become a paradigm for text, vision, and audio tasks (Devlin et al., 2019; Bao et al., 2021; Baevski et al., 2020). In addition to the similarity in the pipeline paradigm, these pre-training models as well have close model structures: On one hand, most of them consist of the following components, ***embedding***, ***encoder***, ***target embedding***, ***decoder***, and ***target***, on the other hand, many modules in above components are shared among models of different modalities. For example, the transformer module (in encoder component) (Vaswani et al., 2017), which

*Corresponding Author
E-mail: alphezhao@tencent.com

arXiv:2212.06385v2 [cs.CL] 11 Jul 2023

Affiliation

As a rule, the affiliation list is located after the author list. Each line containing affiliations' names and other information about the authors (e.g., e-mail addresses) should be labeled as **affiliation**.

Examples:

TencentPretrain: A Scalable and Flexible Toolkit for Pre-training Models of Different Modalities

Zhe Zhao[1*], Yudong Li[2], Cheng Hou[1], Jing Zhao[1], Rong Tian[1], Weijie Liu[1], Yiren Chen[1], Ningyuan Sun[1], Haoyan Liu[1], Weiquan Mao[1], Han Guo[1], Weigang Guo[1], Taiqiang Wu[1], Tao Zhu[1], Wenhang Shi[3], Chen Chen[1], Shan Huang[1], Sihong Chen[1], Liqun Liu[1], Feifei Li[1], Xiaoshuai Chen[1], Xingwu Sun[1], Zhanhui Kang[1], Xiaoyong Du[3], Linlin Shen[2], Kimmo Yan[1]

[1] Tencent AI Lab
[2] School of Computer Science and Software Engineering, Shenzhen University
[3] School of Information and DEKE, MOE, Renmin University of China

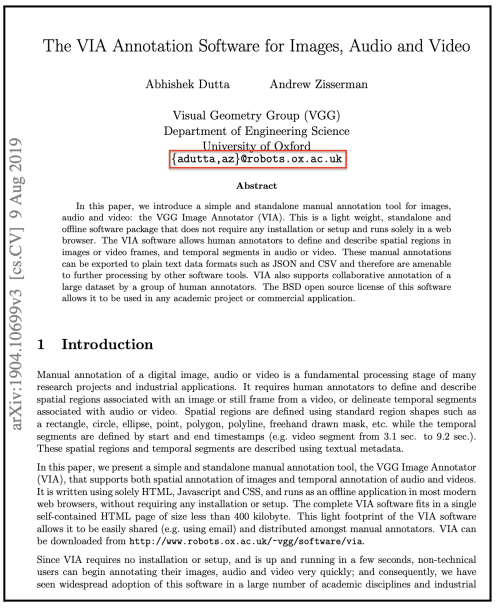
Abstract

Recently, the success of pre-training in text domain has been fully extended to vision, audio, and cross-modal scenarios. The proposed pre-training models of different modalities are showing a rising trend of homogeneity in their model structures, which brings the opportunity to implement different pre-training models within a uniform framework. In this paper, we present TencentPretrain, a toolkit supporting pre-training models of different modalities. The core feature of TencentPretrain is the modular design. The toolkit uniformly divides pre-training models into 5 components: ***embedding***, ***encoder***, ***target embedding***, ***decoder***, and ***target***. As almost all of common modules are provided in each component, users can choose the desired modules from different components to build a complete pre-training model. The modular design enables users to efficiently reproduce existing pre-training models or build brand-new one. We test the toolkit on text, vision, and audio benchmarks and show that it can match the performance of the original implementations.

is successful in the field of text, is increasingly being applied to the vision and audio modalities. (Dosovitskiy et al., 2020; Gulati et al., 2020). Table 1 lists the commonly used pre-training models and their modules.

The trend towards homogeneity in pre-training models is becoming more apparent, which makes it possible to integrate them into a uniform framework. A representative work in this direction is Huggingface Transformers (Wolf et al., 2020), which exploits a non-modular design mode. For each pre-training model in Huggingface Transformers, several separate classes are created, and the code is not refactored with additional abstractions. Users can develop their pre-training models independently which is useful to collaborative development in the community. However, in this design mode, users need to implement the model from scratch when adding a new pre-training model, requiring considerable code work. In addition, with the increased number of pre-training models, the number of classes and lines of code also increases linearly. Codes with the same function may be written many times, which degrades the readability and maintainability of the project.

1 Introduction



The VIA Annotation Software for Images, Audio and Video

Abhishek Dutta Andrew Zisserman

Visual Geometry Group (VGG)
Department of Engineering Science
University of Oxford
{adutta,az}@robots.ox.ac.uk

Abstract

In this paper, we introduce a simple and standalone manual annotation tool for images, audio and video: the VGG Image Annotator (VIA). This is a light weight, standalone and offline software package that does not require any installation or setup and runs solely in a web browser. The VIA software allows human annotators to define and describe spatial regions in images or video frames, and temporal segments in audio or video. These manual annotations can be exported to plain text data formats such as JSON and CSV and therefore are amenable to further processing by other software tools. VIA also supports collaborative annotation of a large dataset by a group of human annotators. The BSD open source license of this software allows it to be used in any academic project or commercial application.

1 Introduction

Manual annotation of a digital image, audio or video is a fundamental processing stage of many research projects and industrial applications. It requires human annotators to define and describe spatial regions associated with an image or still frame from a video, or delineate temporal segments associated with audio or video. Spatial regions are defined using standard region shapes such as a rectangle, circle, ellipse, point, polygon, polyline, freehand drawn mask, etc. while the temporal segments are defined by start and end timestamps (e.g. video segment from 3.1 sec. to 9.2 sec.). These spatial regions and temporal segments are described using textual metadata.

In this paper, we present a simple and standalone manual annotation tool, the VGG Image Annotator (VIA), that supports both spatial annotation of images and temporal annotation of audio and videos. It is written using solely HTML, Javascript and CSS, and runs as an offline application in most modern web browsers, without requiring any installation or setup. The complete VIA software fits in a single self-contained HTML page of size less than 400 kilobyte. This light footprint of the VIA software allows it to be easily shared (e.g. using email) and distributed amongst manual annotators. VIA can be downloaded from http://www.robots.ox.ac.uk/~vgg/software/via.

Since VIA requires no installation or setup, and is up and running in a few seconds, non-technical users can begin annotating their images, audio and video very quickly; and consequently, we have seen widespread adoption of this software in a large number of academic disciplines and industrial

Named item

Named item is a title of some structure part of the article: section, subsection, etc. The line should be labeled as **named_item** if it contains **only text of the title**. If there is additional raw text (not title), then the line should be labeled as **raw_text**.

Examples:

The VIA Annotation Software for Images, Audio and Video

Abhishek Dutta Andrew Zisserman

Visual Geometry Group (VGG)
Department of Engineering Science
University of Oxford
{adutta,az}@robots.ox.ac.uk

Abstract

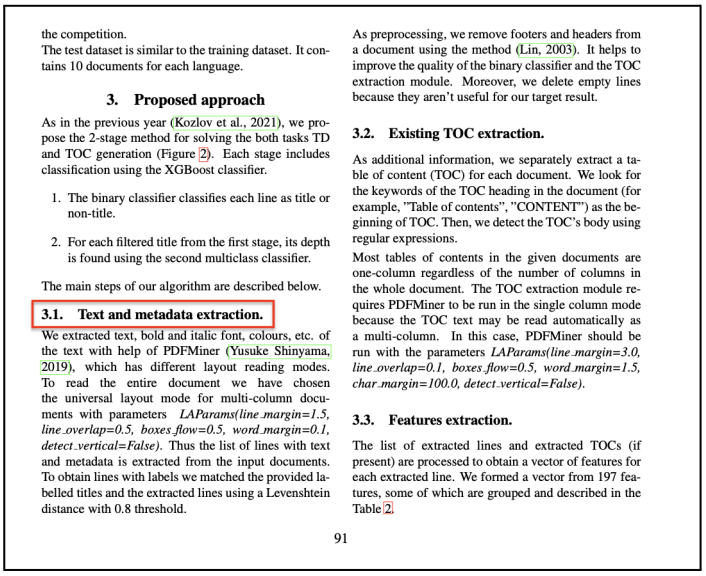
In this paper, we introduce a simple and standalone manual annotation tool for images, audio and video: the VGG Image Annotator (VIA). This is a light weight, standalone and offline software package that does not require any installation or setup and runs solely in a web browser. The VIA software allows human annotators to define and describe spatial regions in images or video frames, and temporal segments in audio or video. These manual annotations can be exported to plain text data formats such as JSON and CSV and therefore are amenable to further processing by other software tools. VIA also supports collaborative annotation of a large dataset by a group of human annotators. The BSD open source license of this software allows it to be used in any academic project or commercial application.

1 Introduction

Manual annotation of a digital image, audio or video is a fundamental processing stage of many research projects and industrial applications. It requires human annotators to define and describe spatial regions associated with an image or still frame from a video, or delineate temporal segments associated with audio or video. Spatial regions are defined using standard region shapes such as a rectangle, circle, ellipse, point, polygon, polyline, freehand drawn mask, etc. while the temporal segments are defined by start and end timestamps (e.g. video segment from 3.1 sec. to 9.2 sec.). These spatial regions and temporal segments are described using textual metadata.

In this paper, we present a simple and standalone manual annotation tool, the VGG Image Annotator (VIA), that supports both spatial annotation of images and temporal annotation of audio and videos. It is written using solely HTML, Javascript and CSS, and runs as an offline application in most modern web browsers, without requiring any installation or setup. The complete VIA software fits in a single self-contained HTML page of size less than 400 kilobyte. This light footprint of the VIA software allows it to be easily shared (e.g. using email) and distributed amongst manual annotators. VIA can be downloaded from http://www.robots.ox.ac.uk/~vgg/software/via.

Since VIA requires no installation or setup, and is up and running in a few seconds, non-technical users can begin annotating their images, audio and video very quickly; and consequently, we have seen widespread adoption of this software in a large number of academic disciplines and industrial



the competition.
The test dataset is similar to the training dataset. It contains 10 documents for each language.

3. Proposed approach

As in the previous year (Kozlov et al., 2021), we propose the 2-stage method for solving the both tasks TD and TOC generation (Figure 2). Each stage includes classification using the XGBoost classifier.

1. The binary classifier classifies each line as title or non-title.
2. For each filtered title from the first stage, its depth is found using the second multiclass classifier.

The main steps of our algorithm are described below.

3.1. Text and metadata extraction.

We extracted text, bold and italic font, colours, etc. of the text with help of PDFMiner (Yusuke Shinyama, 2019), which has different layout reading modes. To read the entire document we have chosen the universal layout mode for multi-column documents with parameters *LAParams(line_margin=1.5, line_overlap=0.5, boxes_flow=0.5, word_margin=0.1, detect_vertical=False)*. Thus the list of lines with text and metadata is extracted from the input documents. To obtain lines with labels we matched the provided labelled titles and the extracted lines using a Levenshtein distance with 0.8 threshold.

As preprocessing, we remove footers and headers from a document using the method (Lin, 2003). It helps to improve the quality of the binary classifier and the TOC extraction module. Moreover, we delete empty lines because they aren't useful for our target result.

3.2. Existing TOC extraction.

As additional information, we separately extract a table of content (TOC) for each document. We look for the keywords of the TOC heading in the document (for example, "Table of contents", "CONTENT") as the beginning of TOC. Then, we detect the TOC's body using regular expressions.

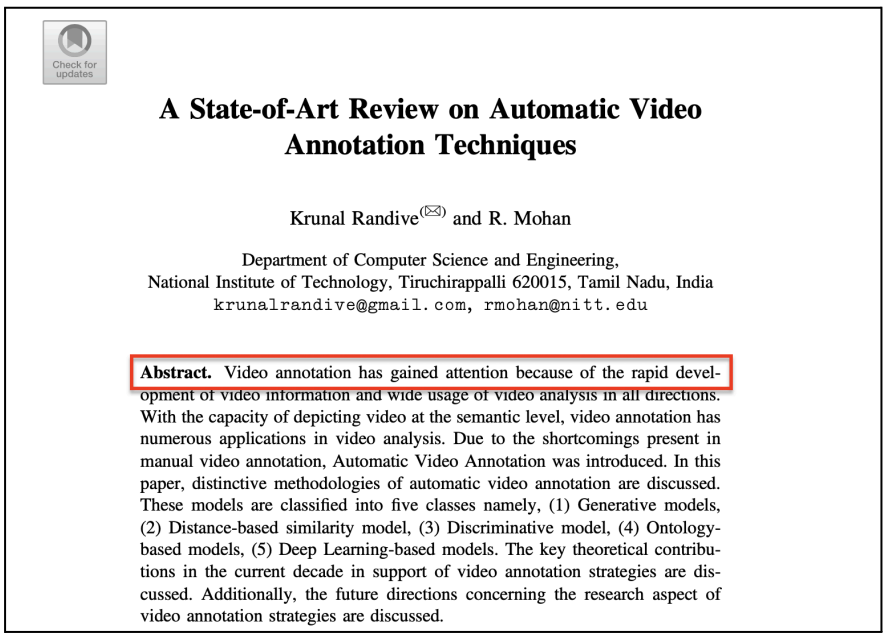
Most tables of contents in the given documents are one-column regardless of the number of columns in the whole document. The TOC extraction module requires PDFMiner to be run in the single column mode because the TOC text may be read automatically as a multi-column. In this case, PDFMiner should be run with the parameters *LAParams(line_margin=3.0, line_overlap=0.1, boxes_flow=0.5, word_margin=1.5, char_margin=100.0, detect_vertical=False)*.

3.3. Features extraction.

The list of extracted lines and extracted TOCs (if present) are processed to obtain a vector of features for each extracted line. We formed a vector from 197 features, some of which are grouped and described in the Table 2.

91

In the example below, the line in the bounding box should be labeled as **raw_text**:



Check for updates

A State-of-Art Review on Automatic Video Annotation Techniques

Krunal Randive(✉) and R. Mohan

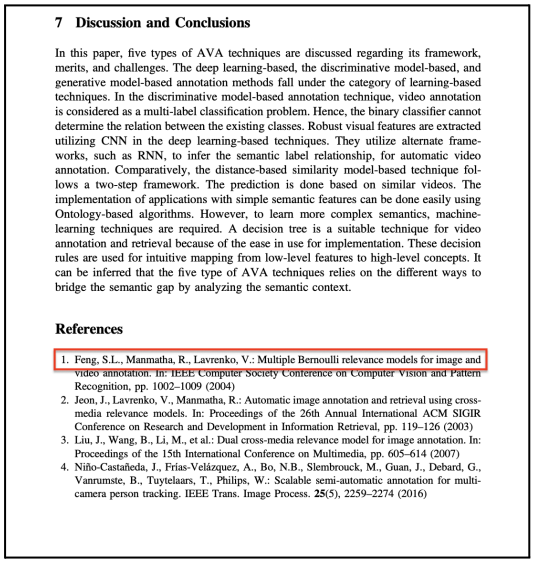
Department of Computer Science and Engineering,
National Institute of Technology, Tiruchirappalli 620015, Tamil Nadu, India
krunalrandive@gmail.com, rmohan@nitt.edu

Abstract. Video annotation has gained attention because of the rapid development of video information and wide usage of video analysis in all directions. With the capacity of depicting video at the semantic level, video annotation has numerous applications in video analysis. Due to the shortcomings present in manual video annotation, Automatic Video Annotation was introduced. In this paper, distinctive methodologies of automatic video annotation are discussed. These models are classified into five classes namely, (1) Generative models, (2) Distance-based similarity model, (3) Discriminative model, (4) Ontology-based models, (5) Deep Learning-based models. The key theoretical contributions in the current decade in support of video annotation strategies are discussed. Additionally, the future directions concerning the research aspect of video annotation strategies are discussed.

Reference

In the end of the article links to the related works are located, i.e. references. **First** line of each reference (in the section References) should be labeled as **reference**.

Examples:



7 Discussion and Conclusions

In this paper, five types of AVA techniques are discussed regarding its framework, merits, and challenges. The deep learning-based, the discriminative model-based, and generative model-based annotation methods fall under the category of learning-based techniques. In the discriminative model-based annotation technique, video annotation is considered as a multi-label classification problem. Hence, the binary classifier cannot determine the relation between the existing classes. Robust visual features are extracted utilizing CNN in the deep learning-based techniques. They utilize alternate frameworks, such as RNN, to infer the semantic label relationship, for automatic video annotation. Comparatively, the distance-based similarity model-based technique follows a two-step framework. The prediction is done based on similar videos. The implementation of applications with simple semantic features can be done easily using Ontology-based algorithms. However, to learn more complex semantics, machine-learning techniques are required. A decision tree is a suitable technique for video annotation and retrieval because of the ease in use for implementation. These decision rules are used for intuitive mapping from low-level features to high-level concepts. It can be inferred that the five type of AVA techniques relies on the different ways to bridge the semantic gap by analyzing the semantic context.

References

1. Feng, S.L., Manmatha, R., Lavrenko, V.: Multiple Bernoulli relevance models for image and video annotation. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1002–1009 (2004)
2. Jeon, J., Lavrenko, V., Manmatha, R.: Automatic image annotation and retrieval using cross-media relevance models. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 119–126 (2003)
3. Liu, J., Wang, B., Li, M., et al.: Dual cross-media relevance model for image annotation. In: Proceedings of the 15th International Conference on Multimedia, pp. 605–614 (2007)
4. Niño-Castañeda, J., Frías-Velázquez, A., Bo, N.B., Slembrouck, M., Guan, J., Debard, G., Vanrumste, B., Tuytelaars, T., Philips, W.: Scalable semi-automatic annotation for multi-camera person tracking. IEEE Trans. Image Process. **25**(5), 2259–2274 (2016)

2. Plugins: State-of-the-art computer vision models are becoming very accurate for tasks such as segmenting and tracking objects, reading text, detecting keypoints on a human body, and many other tasks commonly assigned to human annotators. These computer vision models can help speed up the manual annotation process by seeding an image with automatically annotated regions and then letting human annotators edit or update these regions to create the final annotation. Thanks to projects like TensorFlow.js, it is now possible to run many of these models in a web browser. We envisage such computer vision models attached as plugins to VIA and running in the background to assist human annotators. We are currently developing such a plugin to automatically track an object in a video.

Acknowledgements: This work is funded by EPSRC programme grant Seebibyte: Visual Search for the Era of Big Data (EP/M013774/1).

References

[1] BigParticle.Cloud. How-to: Generate primary object masks. `https://www.bigparticle.cloud/index.php/how-to-generate-primary-object-masks/`, 2018. Accessed: Mar 2019.

[2] Matilde Malaspina. *15th-century printed Italian editions of Aesopian texts.* PhD thesis, University of Oxford, 2018.

[3] Julia Brasch, Kerry M Goodman, Alex J Noble, Micah Rapp, Seetha Mannepalli, Fabiana Bahna, Venkata P Dandey, Tristan Bepler, Bonnie Berger, Tom Maniatis, Clinton S Potter, Bridget Carragher, Barry Honig, and Lawrence Shapiro. Visualization of clustered protocadherin neuronal self-recognition complexes. volume 569, page 280. Nature Publishing Group, 2019.

[4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

Caption

Above or below the article's figures, tables, charts, etc. captions are located. They may contain descriptions, or only numeration and a title. Each line containing a text of the caption should be labeled as **caption**.

Examples:

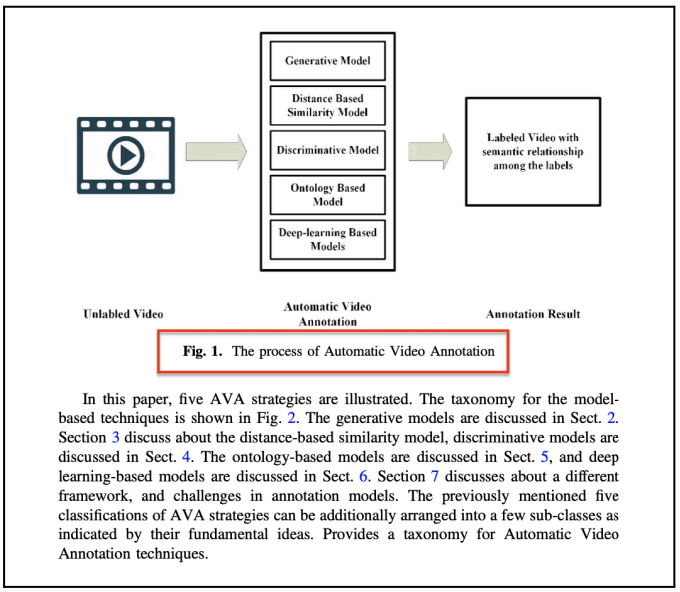
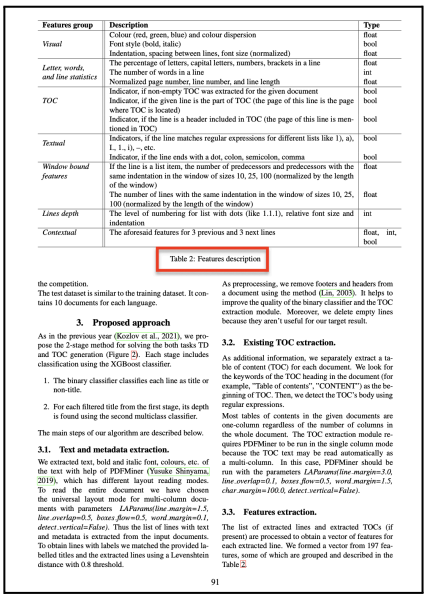


Fig. 1. The process of Automatic Video Annotation

In this paper, five AVA strategies are illustrated. The taxonomy for the model-based techniques is shown in Fig. 2. The generative models are discussed in Sect. 2. Section 3 discuss about the distance-based similarity model, discriminative models are discussed in Sect. 4. The ontology-based models are discussed in Sect. 5, and deep learning-based models are discussed in Sect. 6. Section 7 discusses about a different framework, and challenges in annotation models. The previously mentioned five classifications of AVA strategies can be additionally arranged into a few sub-classes as indicated by their fundamental ideas. Provides a taxonomy for Automatic Video Annotation techniques.



Features group	Description	Type
Visual	Colour (red, green, blue) and colour dispersion	float
	Font style (bold, italic)	bool
	Indentation, spacing between lines, font size (normalized)	float
Letter, words, and line statistics	The percentage of letters, capital letters, numbers, brackets in a line	float
	The number of words in a line	int
	Normalized page number, line number, and line length	float
TOC	Indicator, if non-empty TOC was extracted for the given document	bool
	Indicator, if the given line is the part of TOC (the page of this line is the page where TOC is located)	bool
	Indicator, if the line is a header included in TOC (the page of this line is mentioned in TOC)	bool
Textual	Indicators, if the line matches regular expressions for different lists like 1), a), I., 1., i), –, etc.	bool
	Indicator, if the line ends with a dot, colon, semicolon, comma	bool
Window bound features	If the line is a list item, the number of predecessors and predecessors with the same indentation in the window of sizes 10, 25, 100 (normalized by the length of the window)	float
	The number of lines with the same indentation in the window of sizes 10, 25, 100 (normalized by the length of the window)	float
Lines depth	The level of numbering for list with dots (like 1.1.1), relative font size and indentation	int
Contextual	The aforesaid features for 3 previous and 3 next lines	float, int, bool

Table 2: Features description

the competition.
The test dataset is similar to the training dataset. It contains 10 documents for each language.

3. Proposed approach

As in the previous year (Kozlov et al., 2021), we propose the 2-stage method for solving the both tasks TD and TOC generation (Figure 2). Each stage includes classification using the XGBoost classifier.

1. The binary classifier classifies each line as title or non-title.
2. For each filtered title from the first stage, its depth is found using the second multiclass classifier.

The main steps of our algorithm are described below.

3.1. Text and metadata extraction.

We extracted text, bold and italic font, colours, etc. of the text with help of PDFMiner (Yusuke Shinyama, 2019), which has different layout reading modes. To read the entire document we have chosen the universal layout mode for multi-column documents with parameters *LAParams(line_margin=1.5, line_overlap=0.5, boxes_flow=0.5, word_margin=0.1, detect_vertical=False)*. Thus the list of lines with text and metadata is extracted from the input documents. To obtain lines with labels we matched the provided labelled titles and the extracted lines using a Levenshtein distance with 0.8 threshold.

As preprocessing, we remove footers and headers from a document using the method (Lin, 2003). It helps to improve the quality of the binary classifier and the TOC extraction module. Moreover, we delete empty lines because they aren't useful for our target result.

3.2. Existing TOC extraction.

As additional information, we separately extract a table of content (TOC) for each document. We look for the keywords of the TOC heading in the document (for example, "Table of contents", "CONTENT") as the beginning of TOC. Then, we detect the TOC's body using regular expressions.
Most tables of contents in the given documents are one-column regardless of the number of columns in the whole document. The TOC extraction module requires PDFMiner to be run in the single column mode because the TOC text may be read automatically as a multi-column. In this case, PDFMiner should be run with the parameters *LAParams(line_margin=3.0, line_overlap=0.1, boxes_flow=0.5, word_margin=1.5, char_margin=100.0, detect_vertical=False)*.

3.3. Features extraction.

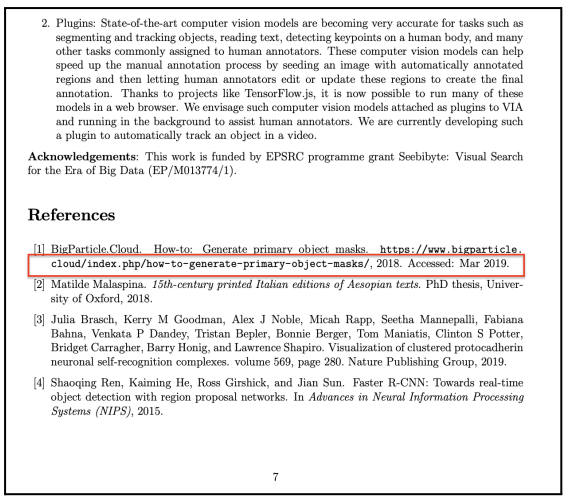
The list of extracted lines and extracted TOCs (if present) are processed to obtain a vector of features for each extracted line. We formed a vector from 197 features, some of which are grouped and described in the Table 2.

91

Raw text

Each line containing a text of the article (except the aforesaid types) should be labeled as **raw_text**. Footnotes are also considered as a raw text.

Examples:



2. Plugins: State-of-the-art computer vision models are becoming very accurate for tasks such as segmenting and tracking objects, reading text, detecting keypoints on a human body, and many other tasks commonly assigned to human annotators. These computer vision models can help speed up the manual annotation process by seeding an image with automatically annotated regions and then letting human annotators edit or update these regions to create the final annotation. Thanks to projects like TensorFlow.js, it is now possible to run many of these models in a web browser. We envisage such computer vision models attached as plugins to VIA and running in the background to assist human annotators. We are currently developing such a plugin to automatically track an object in a video.

Acknowledgements: This work is funded by EPSRC programme grant Seebibyte: Visual Search for the Era of Big Data (EP/M013774/1).

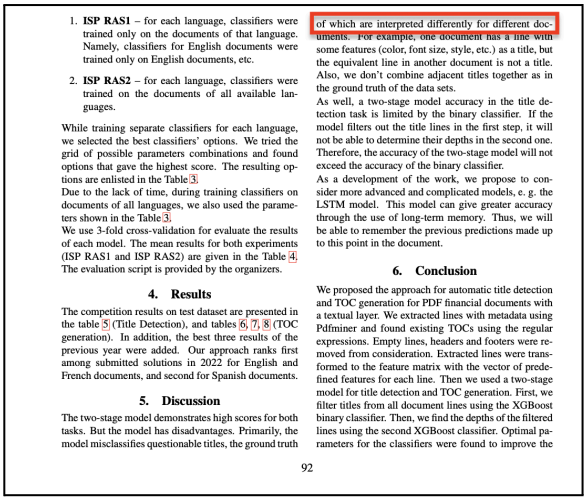
References

[1] BigParticle.Cloud. How-to: Generate primary object masks. https://www.bigparticle.cloud/index.php/how-to-generate-primary-object-masks/, 2018. Accessed: Mar 2019.

[2] Matilde Malaspina. *15th-century printed Italian editions of Aesopian texts.* PhD thesis, University of Oxford, 2018.

[3] Julia Brasch, Kerry M Goodman, Alex J Noble, Micah Rapp, Seetha Mannepalli, Fabiana Bahna, Venkata P Dandey, Tristan Bepler, Bonnie Berger, Tom Maniatis, Clinton S Potter, Bridget Carragher, Barry Honig, and Lawrence Shapiro. Visualization of clustered protocadherin neuronal self-recognition complexes. volume 569, page 280. Nature Publishing Group, 2019.

[4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.



1. **ISP RAS1** – for each language, classifiers were trained only on the documents of that language. Namely, classifiers for English documents were trained only on English documents, etc.

2. **ISP RAS2** – for each language, classifiers were trained on the documents of all available languages.

While training separate classifiers for each language, we selected the best classifiers' options. We tried the grid of possible parameters combinations and found options that gave the highest score. The resulting options are enlisted in the Table 3.
Due to the lack of time, during training classifiers on documents of all languages, we also used the parameters shown in the Table 3.
We use 3-fold cross-validation for evaluate the results of each model. The mean results for both experiments (ISP RAS1 and ISP RAS2) are given in the Table 4.
The evaluation script is provided by the organizers.

4. Results

The competition results on test dataset are presented in the table 5 (Title Detection), and tables 6, 7, 8 (TOC generation). In addition, the best three results of the previous year were added. Our approach ranks first among submitted solutions in 2022 for English and French documents, and second for Spanish documents.

5. Discussion

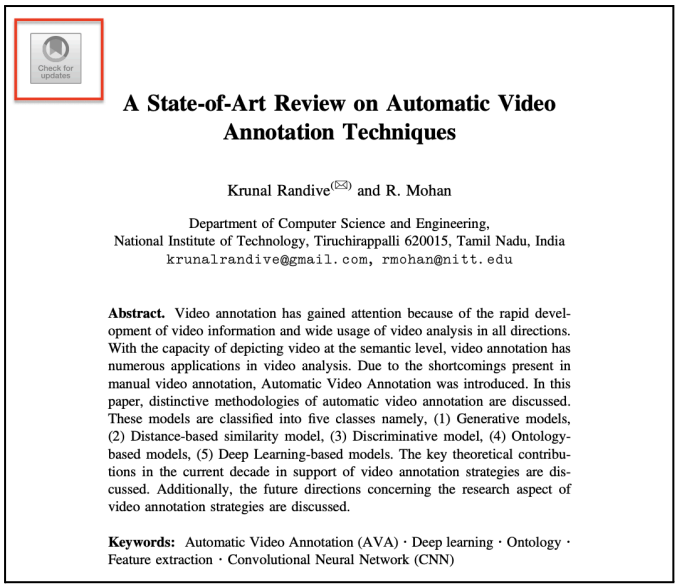
The two-stage model demonstrates high scores for both tasks. But the model has disadvantages. Primarily, the model misclassifies questionable titles, the ground truth of which are interpreted differently for different documents. For example, one document has a line with some features (color, font size, style, etc.) as a title, but the equivalent line in another document is not a title. Also, we don't combine adjacent titles together as in the ground truth of the data sets.
As well, a two-stage model accuracy in the title detection task is limited by the binary classifier. If the model filters out the title lines in the first step, it will not be able to determine their depths in the second one. Therefore, the accuracy of the two-stage model will not exceed the accuracy of the binary classifier.
As a development of the work, we propose to consider more advanced and complicated models, e. g. the LSTM model. This model can give greater accuracy through the use of long-term memory. Thus, we will be able to remember the previous predictions made up to this point in the document.

6. Conclusion

We proposed the approach for automatic title detection and TOC generation for PDF financial documents with a textual layer. We extracted lines with metadata using Pdfminer and found existing TOCs using the regular expressions. Empty lines, headers and footers were removed from consideration. Extracted lines were transformed to the feature matrix with the vector of predefined features for each line. Then we used a two-stage model for title detection and TOC generation. First, we filter titles from all document lines using the XGBoost binary classifier. Then, we find the depths of the filtered lines using the second XGBoost classifier. Optimal parameters for the classifiers were found to improve the

Other

If there is non-textual content in the bounding box, or text from the article's chart, page number, or vertical text near the document borders, it should be labeled as **other**.
Examples:



Check for updates

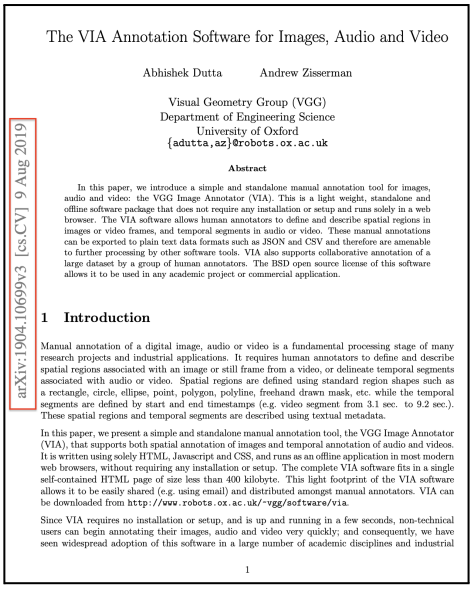
A State-of-Art Review on Automatic Video Annotation Techniques

Krunal Randive(✉) and R. Mohan

Department of Computer Science and Engineering,
National Institute of Technology, Tiruchirappalli 620015, Tamil Nadu, India
krunalrandive@gmail.com, rmohan@nitt.edu

Abstract. Video annotation has gained attention because of the rapid development of video information and wide usage of video analysis in all directions. With the capacity of depicting video at the semantic level, video annotation has numerous applications in video analysis. Due to the shortcomings present in manual video annotation, Automatic Video Annotation was introduced. In this paper, distinctive methodologies of automatic video annotation are discussed. These models are classified into five classes namely, (1) Generative models, (2) Distance-based similarity model, (3) Discriminative model, (4) Ontology-based models, (5) Deep Learning-based models. The key theoretical contributions in the current decade in support of video annotation strategies are discussed. Additionally, the future directions concerning the research aspect of video annotation strategies are discussed.

Keywords: Automatic Video Annotation (AVA) · Deep learning · Ontology · Feature extraction · Convolutional Neural Network (CNN)



The VIA Annotation Software for Images, Audio and Video

Abhishek Dutta Andrew Zisserman

Visual Geometry Group (VGG)
Department of Engineering Science
University of Oxford
{adutta,az}@robots.ox.ac.uk

Abstract

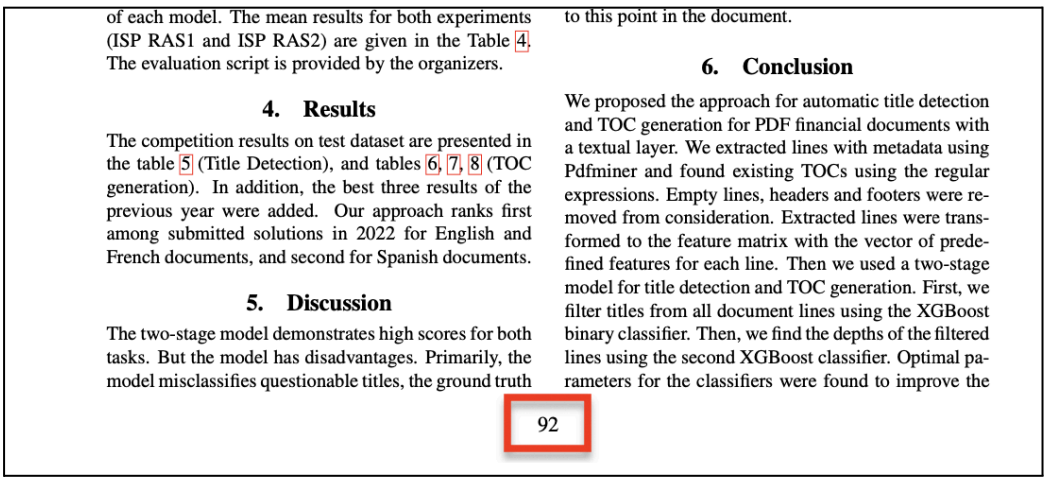
In this paper, we introduce a simple and standalone manual annotation tool for images, audio and video: the VGG Image Annotator (VIA). This is a light weight, standalone and offline software package that does not require any installation or setup and runs solely in a web browser. The VIA software allows human annotators to define and describe spatial regions in images or video frames, and temporal segments in audio or video. These manual annotations can be exported to plain text data formats such as JSON and CSV and therefore are amenable to further processing by other software tools. VIA also supports collaborative annotation of a large dataset by a group of human annotators. The BSD open source license of this software allows it to be used in any academic project or commercial application.

1 Introduction

Manual annotation of a digital image, audio or video is a fundamental processing stage of many research projects and industrial applications. It requires human annotators to define and describe spatial regions associated with an image or still frame from a video, or delineate temporal segments associated with audio or video. Spatial regions are defined using standard region shapes such as a rectangle, circle, ellipse, point, polygon, polyline, freehand drawn mask, etc. while the temporal segments are defined by start and end timestamps (e.g. video segment from 3.1 sec. to 9.2 sec.). These spatial regions and temporal segments are described using textual metadata.

In this paper, we present a simple and standalone manual annotation tool, the VGG Image Annotator (VIA), that supports both spatial annotation of images and temporal annotation of audio and videos. It is written using solely HTML, Javascript and CSS, and runs as an offline application in most modern web browsers, without requiring any installation or setup. The complete VIA software fits in a single self-contained HTML page of size less than 400 kilobyte. This light footprint of the VIA software allows it to be easily shared (e.g. using email) and distributed amongst manual annotators. VIA can be downloaded from http://www.robots.ox.ac.uk/~vgg/software/via.

Since VIA requires no installation or setup, and is up and running in a few seconds, non-technical users can begin annotating their images, audio and video very quickly; and consequently, we have seen widespread adoption of this software in a large number of academic disciplines and industrial



of each model. The mean results for both experiments (ISP RAS1 and ISP RAS2) are given in the Table 4. The evaluation script is provided by the organizers.

4. Results

The competition results on test dataset are presented in the table 5 (Title Detection), and tables 6, 7, 8 (TOC generation). In addition, the best three results of the previous year were added. Our approach ranks first among submitted solutions in 2022 for English and French documents, and second for Spanish documents.

5. Discussion

The two-stage model demonstrates high scores for both tasks. But the model has disadvantages. Primarily, the model misclassifies questionable titles, the ground truth to this point in the document.

6. Conclusion

We proposed the approach for automatic title detection and TOC generation for PDF financial documents with a textual layer. We extracted lines with metadata using Pdfminer and found existing TOCs using the regular expressions. Empty lines, headers and footers were removed from consideration. Extracted lines were transformed to the feature matrix with the vector of predefined features for each line. Then we used a two-stage model for title detection and TOC generation. First, we filter titles from all document lines using the XGBoost binary classifier. Then, we find the depths of the filtered lines using the second XGBoost classifier. Optimal parameters for the classifiers were found to improve the

92