Metadata of the chapter that will be visualized in SpringerLink

Book Title	Proceedings of the Tenth International Conference on Soft Computing and Pattern Recognition (SoCPaR 2018)
Series Title	
Chapter Title	Improving Audiovisual Content Annotation Through a Semi-automated Process Based on Deep Learning
Copyright Year	2020
Copyright HolderName	Springer Nature Switzerland AG

Author	Family Name	**Vilaça**
	Particle	
	Given Name	**Luís**
	Prefix	
	Suffix	
	Role	
	Division	School of Engineering
	Organization	Polytechnic of Porto
	Address	Porto, Portugal
	Email	1121405@isep.ipp.pt
Corresponding Author	Family Name	**Viana**
	Particle	
	Given Name	**Paula**
	Prefix	
	Suffix	
	Role	
	Division	School of Engineering
	Organization	Polytechnic of Porto
	Address	Porto, Portugal
	Division	
	Organization	INESC TEC
	Address	Porto, Portugal
	Email	paula.viana@inesctec.pt
Author	Family Name	**Carvalho**
	Particle	
	Given Name	**Pedro**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	INESC TEC
	Address	Porto, Portugal
	Email	pedro.carvalho@inesctec.pt
Author	Family Name	**Andrade**
	Particle	

Given Name	**Teresa**
Prefix	
Suffix	
Role	
Division	
Organization	INESC TEC
Address	Porto, Portugal
Division	Faculty of Engineering
Organization	University of Porto
Address	Porto, Portugal
Email	mandrade@inesctec.pt

Abstract

Over the last years, Deep Learning has become one of the most popular research fields of Artificial Intelligence. Several approaches have been developed to address conventional challenges of AI. In computer vision, these methods provide the means to solve tasks like image classification, object identification and extraction of features.

In this paper, some approaches to face detection and recognition are presented and analyzed, in order to identify the one with the best performance. The main objective is to automate the annotation of a large dataset and to avoid the costy and time-consuming process of content annotation. The approach follows the concept of incremental learning and a R-CNN model was implemented. Tests were conducted with the objective of detecting and recognizing one personality within image and video content.

Results coming from this initial automatic process are then made available to an auxiliary tool that enables further validation of the annotations prior to uploading them to the archive.

Tests show that, even with a small size dataset, the results obtained are satisfactory.

Keywords (separated by '-')

Content annotation - Computer Vision - Machine Learning - Deep Learning - Object detection - Facial detection - Facial recognition

Improving Audiovisual Content Annotation Through a Semi-automated Process Based on Deep Learning

Luís Vilaça[1], Paula Viana[1,2](✉), Pedro Carvalho[2], and Teresa Andrade[2,3]

[1] School of Engineering, Polytechnic of Porto, Porto, Portugal
1121405@isep.ipp.pt
[2] INESC TEC, Porto, Portugal
{paula.viana,pedro.carvalho,mandrade}@inesctec.pt
[3] Faculty of Engineering, University of Porto, Porto, Portugal

Abstract. Over the last years, Deep Learning has become one of the most popular research fields of Artificial Intelligence. Several approaches have been developed to address conventional challenges of AI. In computer vision, these methods provide the means to solve tasks like image classification, object identification and extraction of features.

In this paper, some approaches to face detection and recognition are presented and analyzed, in order to identify the one with the best performance. The main objective is to automate the annotation of a large dataset and to avoid the costy and time-consuming process of content annotation. The approach follows the concept of incremental learning and a R-CNN model was implemented. Tests were conducted with the objective of detecting and recognizing one personality within image and video content.

Results coming from this initial automatic process are then made available to an auxiliary tool that enables further validation of the annotations prior to uploading them to the archive.

Tests show that, even with a small size dataset, the results obtained are satisfactory.

Keywords: Content annotation · Computer Vision · Machine Learning · Deep Learning · Object detection · Facial detection · Facial recognition

1 Introduction

Machine Learning (ML) has been applied in problems that require a high degree of manual parametrization or the manipulation of large sets of data. When combined with Computer Vision (CV), where the goal is to extract useful information from digital images, ML has demonstrated good results when comparing with traditional algorithms. Object detection, facial detection and recognition, are examples of problems that have been addressed.

A. M. Madureira et al. (Eds.): SoCPaR 2018, AISC 942, pp. 1–10, 2020.
https://doi.org/10.1007/978-3-030-17065-3_7

The main objective of this paper is to semi-automate the process of image and video content annotation, by using ML algorithms for face detection and recognition. In this first version, a single person of interest was selected. One of the restrictions for the solution to be implemented is that the data set used for training should be small to avoid the need of manually annotation the content for creating the ground truth.

The proposed solution creates a pipeline of two applications: an application for automatically generating the pre-annotations after an initial short training phase and a validation application enabling cleaning any noisy annotations and creating valid information to be further used to improve the accuracy.

The remaining of this paper is structured as follows. A brief review of the related work is presented in Sect. 2. The experiments, evaluation parameters, results and decision taking is presented in Sect. 3. Section 4 presents the proposed solution, along with its functionalities. Finally, the final section concludes this document with some discussion and conclusions.

2 Related Work

Searching and browsing large collections of video assets depends greatly on the capacity of describing this content. Several approaches have been proposed in the literature to enhance the accuracy of search queries. Examples based on image, video, audio or text analysis as well as on semantic related approaches can be found applied to several areas of application [3,17]. Other methods have been exploiting user contributions and implementing methods that automatically filter noisy information [14,19,20,30].

Approaches based on image processing date from mid-1960s and have been evolving progressively. In 2001, Viola and Jones developed a facial detection algorithm, later called "Viola-Jones Algorithm" [31], that is usually regarded as an important milestone. It was the first time a facial detection software showed acceptable results for real-time analysis. Some facial features, such as the eyes, mouth, nose, and the relationships between them, were manually coded and then, the information was submitted in a binary classifier, with the purpose of classifying the face as true or false. This algorithm was distinguished by its fast detection capability when compared to SoA solutions. Its main limitation was the decrease of accuracy with the non-frontality of the faces.

In 2005 Dadal and Triggs developed a more efficient technique the "Histograms of Oriented Gradients" [5]. This algorithm, still used nowadays, was applied to face detection as well as to other problems such as pedestrians detection. Pixels in the image are compared to its neighbors, enabling the detection of maximum gradients. Classification was achieved by comparison with pre-coded facial features and the decision was based on a confidence threshold.

Deep Learning (DL) matures in 2010 assuming the neural networks, a computational method inspired by the brains biological networks, as its core. In the same year, at ImageNet, Krizhevsky et al. used a Convolutional Neural Network (CNN) [12] that outperformed all other algorithms. This was possible due to the amount of audiovisual information made available for training and to the ever-increasing graphic processing power.

Although these results are promising, they still have some drawback that limit their application to real use cases as images used for training were preprocessed and only contained the target objects. Moreover, this mechanism is quite computationally demanding and trying to overcome this problem, in 2015, the concept of a Regional Convolutional Neural Networks (R-CNN) [7] was proposed. This approach uses a selective search process to create windows of different sizes within the original image and each one groups adjacent pixels with similar characteristics ("Region Proposals"). After this initial pre-processing phase, each window is inserted into a trained CNN for classification. Several adaptation approaches have been developed to improve the efficiency of this method: Fast R-CNN [6], Faster R-CNN [23] and Mask R-CNN [8].

Although this learning approach provides good results, it relies heavy on the training sets. Alternative techniques optimized for facial detection and recognition, that might not rely on learning processes, have also been exploited.

Four main categories can be identified for facial detection: (1) Knowledge-based models - use hand coded features, defined as rules from the human knowledge of what best characterize a face; (2) Feature-based approaches - Aims to find structural features that exist even with the variation of the external conditions; (3) Template matching - Uses standard patterns of the face and detects them through correlation evaluation; (4) Appearance-based models: Through a set of examples, the representative variability of the facial appearance is captured.

Within the Knowledge-based model approach, [32] uses a multiresolution hierarchical method to detect faces that, although did not achieve high detection rates, was used in later works due to its simplicity [11]. In contrast to these methods, the feature-based approaches aim to find invariant features of faces for detection. Sirohey [24] proposed a solution to segment a face from a cluttered background, achieving 80% accuracy on 48 images. It uses an edge map [4] and heuristics for grouping and selecting edges, to find the contour of the face. Recently, methods that combine several facial features, like the one used in [33], which is based on structure, color and geometry, have been proposed.

An example of Template matching was described by Tsukamoto et al. in [27,28]. It uses a qualitative model for face pattern (QMF), dividing the sample image into blocks, estimating the qualitative features for each one and defining a template. In the Appearance-based model approach, the template is learned from the training examples using one of the traditional ML approaches.

Facial recognition approaches can be split into two main classes: (1) Template-based - compute the correlation between the template and the face submitted; (2) Geometry feature-based - analyze the local features and their geometric relationships.

For the Template models for facial recognition, statistical tools are used to compute an adequate set of templates: Support Vector Machines [18,29], Linear Discriminant Analysis [2], Principal Component Analysis [25] and Neural Networks [10,15] have been used for this purpose. For Geometry-based solutions, approaches like the one presented by Cootes et al. [13], that uses a built flexible model for face recognition, can be identified.

3 ML Frameworks and Services for Facial Detection and Recognition

The scientific community as well as the industry have been making available ML frameworks that can be used and adapted for several purposes. This section describes the most relevant cloud-based services and software libraries available to implement facial detection and recognition algorithms. The first set of solutions enable detecting faces and returning their location within the image, usually as a surrounding box (BBOX). The second set of algorithms adds the capability of identifying a person from a set of possible options.

Tests were conducted using the following cloud-based services: Microsoft Azure, Google Vision API, Clarifai and Amazon Rekognition. For the architecture based on local software libraries Tensorflow and YOLO [21,22] (Darknet) were tested. For YOLO, Darkflow [1], the translated version of Darknet for Tensorflow, was used. Two different Tensorflow models were tested - SSD Mobilenet [9,16] and Faster RCNN Inception v2 [23,26], while for the YOLO the Tiny model version was the one used. This was due to limitations on the available hardware used for the training phase.

The goal of this testbed is to identify the most efficient facial recognition and detection services available, taking into account the small size of the training dataset.

3.1 Experiment Setup

Given that the application is expected to be used in several scenarios that include the capture of the content in non-controlled environments that comprise indoor and outdoor situations, different illumination and individual as well a group pictures, one of the aspects under analysis is the robustness of the approach with the non-frontality of the faces. Failure of the detection (false negatives), wrong detections (false positives) and the interval between the minimum positive values and the maximum negative values, so that a threshold can be identified, will be considered. In the cloud-based APIs, due to limitations imposed by the frameworks, available pre-trained networks were used. The only exception was in using the Clarifai web-service for which training with our training dataset was done. For the others, the experiment used the functionality of providing a reference image.

For testing purpose, all the services will be analyzed using the same dataset that includes seven images of Prof. Marcelo Rebelo de Sousa and a video in which he also appears. Several other people are also present in the content and the target appears in different locations and positions. Given the limitations imposed by some of the frameworks, the video was fragmented into frames, taken every second, each being analyzed individually. This resulted in 52 additional images in the test dataset. The training dataset consisted of 393 images of Prof. Marcelo Rebelo de Sousa. Each one was annotated manually to generate the ground truth required for the training process. A Supervised Learning method using

Transfer Learning, where a pre-trained model is reused to detect new classes, was implemented. Figure 1 depicts the improvement in respect to the total loss metric returned by the frameworks and that include several parameters that reflect the error of the model (an example is the difference between the ground truth and the returned BBox location). These results were used to define the number of iterations to be used during the training process in order to guarantee an optimal generalization. Training and testing were performed in the same system with the following characteristics: Intel Core i7 6700K, 4.3 GHz; 16 Gb of Ram, at 2400 MHz; Gigabyte GeForce GTX 1060 6 Gb, 1700 MHz.

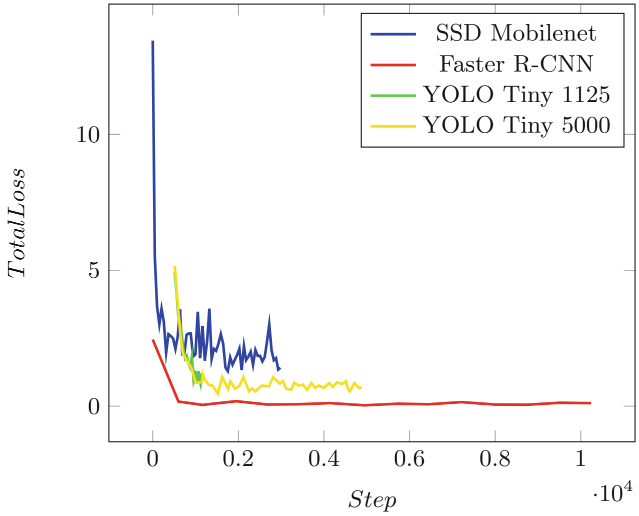


Fig. 1. DL models training evolution

3.2 Results

Figure 2 presents the results obtained by the Cloud-based APIs. Both Clarifai and Google Vision obtained 100% success, locating the target in all the test images. However, it is worth mentioning that among those, Clarifai was the only one that was trained with our training dataset. Additionally, some of these frameworks return extra information providing context information of the image that include image properties, text, emotions and approximate age in facial analysis. Clarifai, Google Vision and Amazon Rekognition provide also semantic aware information returning tags that identify a given scenario. Examples of tags include Sport, People, Parking Lot, Car, Intersection, Urban, Nature, Sunrise and Sunset, Urban Area, Street, Road, City, Skyline, Daytime, Dusk, Evening and Night. Figures 3 and 4 show the results obtained for each of the models of the local frame-works. Testing was performed using the described test dataset.

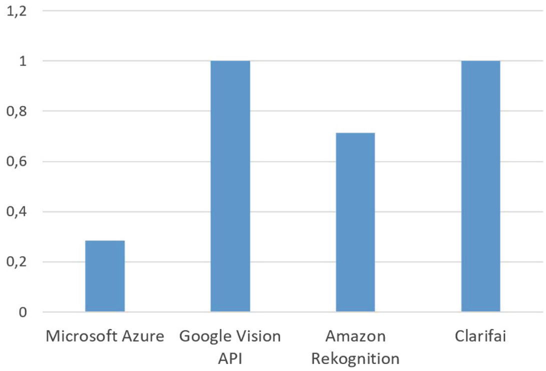


Fig. 2. Detection results with the cloud-based APIs

The results in Fig. 3 present the number of true and false positives for the total number of Bbox generated (one of the test images may contain several Bbox) while in Fig. 4 we can see the true and false positives when having as purpose being able to state if the target person is on an image.

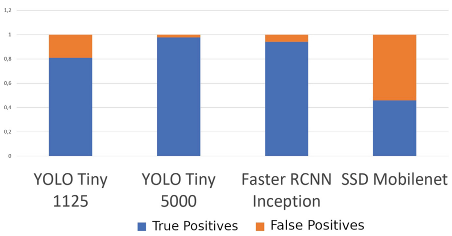


Fig. 3. Percentage of true and false positives per total of BBOXes generated

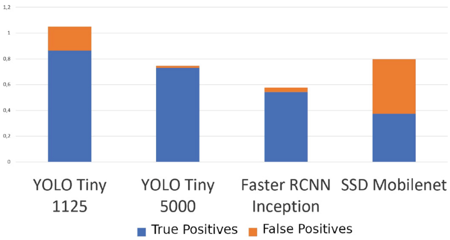


Fig. 4. Percentage of true and false positives per number of images analyzed

3.3 Discussion and Conclusions

The results presented in the previous figures show that the "Faster RCNN Inception" model is the one resulting in more correct identifications. Results have however been achieved at the expense of additional computational costs for the inference process (sec/image): (1) Yolo-Tiny: 0.095; (2) Yolo-Full: 0.287; (3) SSD-Mobilenet: 3.046; (4) Faster-RCNN: 4.366. These results are a direct consequence of the architecture used for each approach and of the pre-processing phase methodology. While in F-RCNN two neural networks are used – one for generating the *region proposals* and the other for classifying those regions – on SSD a neural network is eliminated as pre-processing relies on a random process to generate the initial BBoxes. For the Yolo model, a simpler network enables reducing the processing time and this is further exploited in the Tiny version by reducing the number of layers.

Tests also show that Faster-RCNN returns a higher confidence value than the others, although it still fails to detect 45% of the target person appearing in the images.

By analyzing the results obtained in Fig. 2 and taking into account that Google Vision API does not allow to train a network with a local dataset, the best choice of implementation is with the Clarifai Web Service. This API will be implemented with the general-purpose network to generate tags about the context of the image.

4 Proposed Solution

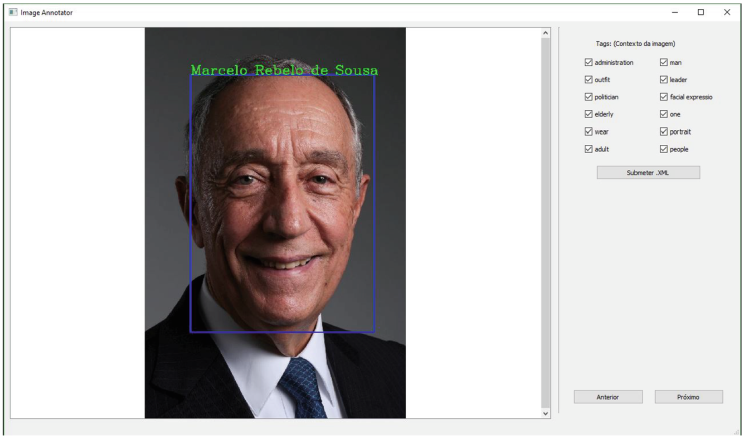
The main goal, as stated before, is to speed up the annotation process of an audiovisual archive. The proposed solution includes a two-stage process that was implemented as two independent software applications developed in Python: the Annotation Generator and the Annotation Validator.

Annotation Generator: This application uses the model selected in the previous chapter to perform the detection of Prof. Marcelo Rebelo de Sousa. A confidence value higher than 60% was defined as a requirement. The same image is submitted to the "General" model of the Clarifai service, to obtain additional tags relative to the image's semantics. The output of this process is a XML file having the structure presented in Listing 1.1.

```
     <annotation>
    <folder>images</folder>
    <filename>image1.jpg</filename>
    <size>
        <width>1619</width>
        <height>1080</height>
    </size>
    <tags>administration, outfit, politician</tags>
    <object>
        <name>Person</name>
        <bndbox>
            <xmin>794</xmin>
            <ymin>41</ymin>
            <xmax>1618</xmax>
            <ymax>1075</ymax>
        </bndbox>
    </object>
</annotation>
```

Listing 1.1. XML File Structure

Annotations resulting from this process are them made available for customization and validation through the Annotation Validator: this application reads the information generated in the previously described step and provides a GUI with editing functionalities. Figure 5 shows the main screen where the tags and the Bbox generated are presented and made available either for validation or for editing. Any action will be reflected on the XML file, updating the descriptions. The full process is depicted in Fig. 6.



Tags: (Contexto da imagem)

☑ administration ☑ man
☑ outfit ☑ leader
☑ politician ☑ facial expressio
☑ elderly ☑ one
☑ wear ☑ portrait
☑ adult ☑ people

Fig. 5. Overview of the UI in the annotation validation application

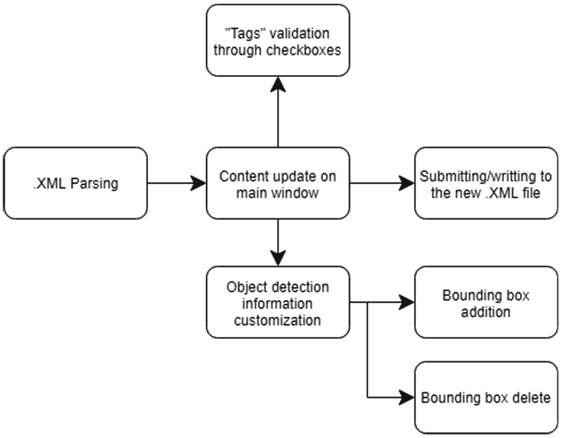

"Tags" validation through checkboxes
.XML Parsing
Content update on main window
Submitting/writting to the new .XML file
Object detection information customization
Bounding box addition
Bounding box delete

Fig. 6. Functionalities of the annotation validation application

5 Conclusions

This paper analyses the performance of several ML approaches for detecting and recognizing a target person considering the small size of the training dataset. A prototype that enables preliminary automatic annotation of the multimedia content followed by a simple and intuitive application that enables removing noisy information or adding additional data is also presented. Initial tests show that the performance of the system enables accelerating the annotation process. The solution that best fits the requirements is the YOLO Tiny and the Faster R-RCNN Inception, this last one outperforming all for the collected dataset. Test showed also that the user validation application could indeed hasten the annotation process while comparing to the fully manual introduction of the same data. Future work includes testing the solution with the YOLO full model and to repeat the tests with training datasets of different sizes.

Acknowledgments. The work presented was partially supported by the following projects: FourEyes, a Research Line within project “TEC4Growth: Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact/NORTE-01- 0145-FEDER-000020” financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF); FotoInMotion funded by H2020 Framework Programme of the European Commission.

References

1. Darkflow repository. https://github.com/thtrieu/darkflow. Accessed 09 July 2018
2. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: recognition using class specific linear projection. Technical report, Yale University, New Haven, United States (1997)
3. Bertini, M., Del Bimbo, A., Torniai, C.: Automatic video annotation using ontologies extended with visual information. In: Proceedings of the 13th Annual ACM International Conference on Multimedia, MULTIMEDIA 2005, pp. 395–398. ACM, New York (2005)
4. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **PAMI-8**(6), 679–698 (1986)
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection, vol. 1, pp. 886–893, June 2005
6. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
7. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. CoRR abs/1311.2524 (2013)
8. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2980–2988. IEEE (2017)
9. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: efficient convolutional neural networks for mobile vision applications. CoRR abs/1704.04861 (2017)
10. Howell, A.J., Buxton, H.: Invariance in radial basis function neural networks in human face classification. Neural Process. Lett. **2**(3), 26–30 (1995)
11. Kotropoulos, C., Pitas, I.: Rule-based face detection in frontal views. In: Proceedings International Conference on Acoustics, Speech and Signal Processing, vol. 4, pp. 2537–2540 (1997)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks, pp. 1097–1105 (2012)
13. Lanitis, A., Taylor, C.J., Cootes, T.F.: Automatic interpretation and coding of face images using flexible models. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7), 743–756 (1997)
14. Larson, M., Soleymani, M., Serdyukov, P., Rudinac, S., Wartena, C., Murdock, V., Friedland, G., Ordelman, R., Jones, G.J.F.: Automatic tagging and geotagging in video collections and communities. In: Proceedings 1st ACM International Conference on Multimedia Retrieval, ICMR 2011, pp. 51:1–51:8. ACM, New York (2011)

15. Lawrence, S., Giles, C.L., Tsoi, A.C., Back, A.D.: Face recognition: a convolutional neural-network approach. IEEE Trans. Neural Netw. **8**(1), 98–113 (1997)
16. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: single shot multibox detector. In: European Conference on Computer Vision, pp. 21–37. Springer (2016)
17. Moxley, E., Mei, T., Hua, X., Ma, W., Manjunath, B.S.: Automatic video annotation through search and mining. In: 2008 IEEE International Conference on Multimedia and Expo, pp. 685–688, June 2008
18. Osuna, E., Freund, R., Girosit, F.: Training support vector machines: an application to face detection. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 130–136, June 1997
19. Pinto, J.P., Viana, P.: TAG4VD: a game for collaborative video annotation. In: Proceedings of the 2013 ACM International Workshop on Immersive Media Experiences, ImmersiveMe 2013, pp. 25–28. ACM, New York (2013)
20. Pinto, J.P., Viana, P.: Using the crowd to boost video annotation processes: a game based approach. In: Proceedings of the 12th European Conference on Visual Media Production, CVMP 2015, pp. 22:1–22:1. ACM, New York (2015)
21. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
22. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. CoRR abs/1612.08242 (2016)
23. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks (2015)
24. Sirohey, S.A.: Human face segmentation and identification. Technical report (1993)
25. Sirovich, L., Kirby, M.: Low-dimensional procedure for the characterization of human faces. J. Opt. Soc. Am. A **4**(3), 519–524 (1987)
26. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. CoRR abs/1512.00567 (2015)
27. Tsukamoto, A., Lee, C.W., Tsuji, S.: Detection and pose estimation of human face with synthesized image models. In: Proceedings of 12th International Conference on Pattern Recognition, vol. 1, pp. 754–757, October 1994
28. Tukamoto, A.: Detection and tracking of human face with synthesized templates. In: Proceedings of the ACCV 1993, pp. 183–186 (1993)
29. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (2013)
30. Viana, P., Pinto, J.P.: A collaborative approach for semantic time-based video annotation using gamification. Hum.-Centric Comput. Inf. Sci. **7**(1), 13 (2017)
31. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features, vol. 1, pp. I-511–I-518 (2001)
32. Yang, G., Huang, T.S.: Human face detection in a complex background. Pattern Recognit. **27**(1), 53–63 (1994)
33. Yang, M.H., Ahuja, N.: Detecting human faces in color images. In: Proceedings of the International Conference on Image Processing, ICIP 1998, vol. 1, pp. 127–130, October 1998

Author Queries

Chapter 7

Query Refs.	**Details Required**	**Author's response**
AQ1	This is to inform you that corresponding author has been identified as per the information available in the Copyright form.	

MARKED PROOF

Please correct and return this set

Please use the proof correction marks shown below for all alterations and corrections. If you wish to return your proof by fax you should ensure that all amendments are written clearly in dark ink and are made well within the page margins.

Instruction to printer	*Textual mark*	*Marginal mark*
Leave unchanged	• • • under matter to remain	Ⓥ
Insert in text the matter indicated in the margin	⋏	New matter followed by ⋏ or ⋏ⓧ2
Delete	/ through single character, rule or underline or ├──┤ through all characters to be deleted	ꝺ or ꝺⓧ2
Substitute character or substitute part of one or more word(s)	/ through letter or ├──┤ through characters	new character / or new characters /
Change to italics	— under matter to be changed	⌙
Change to capitals	≡ under matter to be changed	≡
Change to small capitals	= under matter to be changed	=
Change to bold type	~ under matter to be changed	~
Change to bold italic	≂ under matter to be changed	≂
Change to lower case	Encircle matter to be changed	≢
Change italic to upright type	(As above)	⊥
Change bold to non-bold type	(As above)	⊥
Insert 'superior' character	/ through character or ⋏ where required	Y or ⋏ under character e.g. Y or ⋏
Insert 'inferior' character	(As above)	⋏ over character e.g. ⋏
Insert full stop	(As above)	⊙
Insert comma	(As above)	,
Insert single quotation marks	(As above)	Y or ⋏ and/or Y or ⋏
Insert double quotation marks	(As above)	Y or ⋏ and/or Y or ⋏
Insert hyphen	(As above)	├─┤
Start new paragraph	⌐	⌐
No new paragraph	↩	↩
Transpose	⊔⊓	⊔⊓
Close up	linking ⁐ characters	⁐
Insert or substitute space between characters or words	/ through character or ⋏ where required	Y
Reduce space between characters or words	| between characters or words affected	⊤