*Proc. of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*
*16-18 November 2022, Maldives*

# Exploration of the Vienna City Library Poster Collection using Computer Vision Approaches

Florian Kleber
*Computer Vision Lab TU Wien*
Vienna, Austria
kleber@cvl.tuwien.ac.at

Adam Novozamsky
*Computer Vision Lab TU Wien*
Vienna, Austria
novozamsky@utia.cas.cz

Robert Sablatnig
*Computer Vision Lab TU Wien*
Vienna, Austria
sab@cvl.tuwien.ac.at

Michael Dittenbach
*max.recall information systems*
Vienna, Austria
m.dittenbach@max-recall.com

***Abstract*—Parts of collections of libraries, archives, and e.g. museums can still be uncatalogued. Even if metadata is provided, only standardized information of the described resources (dependent on the collection) is available, i.e., creator names, titles, and subject terms, limiting the search options for experts and typical users. In the case of image-based collections, the information of the image itself can be used as an additional feature to extend the search capabilities of the user.**

**This paper analyzes the use of standard computer vision methods to explore the Vienna City Library poster collection using additional image-based properties. The proposed exploration tool allows a search based on the provided metadata and features based on face detection and retrieval, image retrieval, object detection, text recognition, and main color similarity. The OpenSearch engine is used to index the metadata and visual features, allowing for a real-time search of extensive collections. The qualitative and quantitative analysis shows the potential of visual features within a search tool.**

***Index Terms*—library, collection, search, exploration tool, computer vision**

## I. Introduction

To search in libraries, archives, and so on, researchers and common public users are limited to the indexed metadata of resources. Based on the type of objects (books, pictures, paintings, etc.), additional information like the text itself (books), image features, and objects within an image (paintings, photographs) can be represented in a machine-readable form, indexed, and thus be used for added search capabilities. Due to the number of objects in standard collections, manual annotation of metadata exceeds the budget of public institutions. Based on our experience companies charge about 15$ for a manual transcription compared to about 0.20$ for an automated transcription (e.g. ReadCoop[1]) of a manuscripts page. Thus, state-of-the-art Computer Vision (CV) methodologies can be used to enrich the metadata of collections to facilitate additional search possibilities. This paper proposes a prototype for the Vienna City Library poster collection[2] based on open source CV libraries, which User Interface (UI) is shown in Fig. 1. The UI shows the search options based on the standard metadata of the library (Date, Title, Language, Publisher, etc.), and in addition, the following features are added:

- Face Detection and Retrieval (top right)
- Image Retrieval (above posters on the right)
- Color Search (above posters in the middle)
- OCR (above posters on the left)
- Objects (left sidebar)

The added search options allow experts to retrieve similar posters or posters with specific content quickly. Everyday users (general public) can, e.g., upload a picture of a face and find similar posters in the collection.

The **contribution of this paper** is summarized as follows:

- We provide a *novel* search and exploration tool which can be used in the context of libraries, archives, etc.
- Only open source state-of-the-art CV methodologies are used for the exploration tool of the Vienna City Library, allowing other institutions to use the proposed search tool.
- A quantitative and qualitative assessment is done on parts of the poster collection of the Vienna City Library.
- A demo is presented at https://plakate.cvl.tuwien.ac.at

This paper is structured as follows: In Section II a detailed overview of the poster exploration tool is given, including the search index using OpenSearch II-A. The used state-of-the-art CV methodologies are presented in Section III, including image retrieval III-A, text recognition III-B, face detection and retrieval III-C, object detection III-D and color similarity III-E. Section IV concludes the investigation on the proposed open source search tool.

## II. Poster Exploration Tool

The UI (see Fig. 1) enables a full-text search of poster metadata exported via the OAI (Open Archives Initiative) interface of the ALMA system[3] of the Vienna City Library

The project has been funded by the Wienbibliothek im Rathaus, Magistrat der Stadt Wien, MA9.

[1]https://readcoop.eu/

[2]https://www.wienbibliothek.at/bestaende-sammlungen/plakatsammlung

[3]https://exlibrisgroup.com/products/alma-library-services-platform/

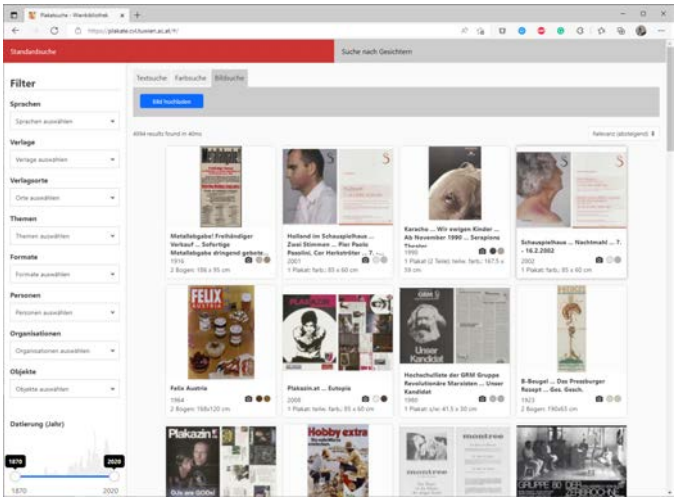

Fig. 1. Interface of the prototype.

and supports an innovative search based on the additionally enriched metadata. This enriched data (e.g., objects, OCR text) is also visualized in the view. The metadata harvested from the Vienna City Library uses the MARCXML (MAchine-Readable Cataloging) data format[4]. Based on this data format, an index scheme was developed, and the following data processing pipeline was realized:

- Harvesting of MARCXML datasets via an OAI interface of the Vienna City Library.
- Metadata importer (MARCXML to index scheme).
- Importer for OCR, objects, faces, images, and color features.

The prototype system uses a representative subset of 5,000 posters chosen by the experts of the Vienna City Library as test data. OpenSearch[5] is used as the technology for indexing the standard and enriched metadata. This variant of Elasticsearch[6] developed by Amazon Web Services has a significant advantage over the standard Elasticsearch developed by the company Elastic, that in addition to the extensive mechanisms for full-text search, a vector-based search based on an efficient k-nearest neighbor (Approximate kNN) implementation is also available as open-source functionality[7], which is essential for the vector-based metadata (e.g., faces) extracted by the CV modules. For the user interface, modern JavaScript-based technologies were used for rapid prototyping: ReactJS Framework[8] as well as ReactiveSearch[9] components for Elasticsearch.

Since the system components are loosely coupled and communicate with each other primarily via HTTP-based interfaces, there are no fundamental restrictions regarding implementation. This can be done on physical servers, virtual machines, container-based, or hybrid. The prototype system runs in Docker containers on a virtual machine and on a server with GPU support. Container management is carried out via a Docker Compose configuration.

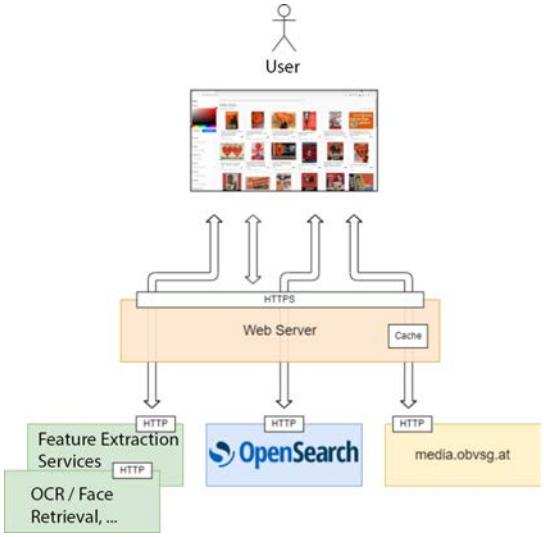


Fig. 2. Prototype architecture of the exploration tool.

The architecture of the prototype is shown in Fig. 2 and includes the following components:

- Web Server: hosts the user interface and acts as a reverse proxy for forwarding requests to the backend services.
- OpenSearch Index: the index serves as the central metadata repository for the posters and is configured as a single node index (no clustering) with authentication and access control disabled, as these are not needed.
- Colour extraction service: Java-based web service that calculates the two most prominent colors for an uploaded image and returns the information in a JSON data structure.
- Face/image feature extraction service: Python-based image processing services for extracting face and general image features are provided via a Flask server. Images uploaded via multipart requests are analyzed, and the corresponding results are returned in a JSON data structure.
- Media server (external): Hosts thumbnails and web versions of posters for display.

### A. *OpenSearch - Data import*

For the prototype, an Apache Camel-based data processing pipeline was developed, which can be controlled via several shell scripts. Apache Camel[10] is an open-source framework for data integration with a variety of connectors. The OAI, file system and Elasticsearch (compatible with OpenSearch) connectors are particularly relevant for this project. With these scripts, it is possible to create the OpenSearch index scheme, load the MARCXML metadata of the posters from the Alma OAI interface and index them, including the metadata resulting

[4]https://www.loc.gov/standards/marcxml/
[5]https://opensearch.org/
[6]https://www.elastic.co/elasticsearch/
[7]In the meantime, the kNN-based search has also been made available in Elasticsearch.
[8]https://reactjs.org/
[9]https://opensource.appbase.io/reactivesearch/
[10]https://camel.apache.org/

from the various CV methodologies. The image files of the posters were processed separately, and the resulting metadata was made available as files in JSON format.

Fig. 3 shows the relationship between the scripts and the processing steps.

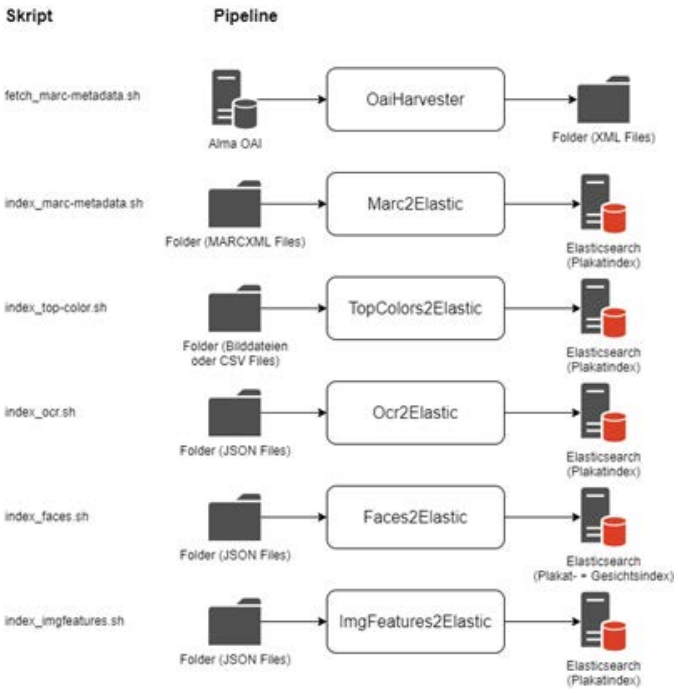


Fig. 3. Processing steps of the exploration tool.

The scripts can be summarized as follows:

- `fetch_marc_metadata.sh` Loads a defined set of poster metadata via the OAI interface from the Alma server and stores it in MARCXML format on the file system.
- `index_marc-metadata.sh` Transforms the MARCXML poster metadata according to the index scheme and sends it to the OpenSearch service for indexing.
- `index_top-color.sh` Reads image files from an input directory, calculates the two most common colors, and stores this information in an output directory as CSV files (one file per image). If the color information for an image has already been calculated, the already generated data is used.
- `index_ocr.sh` Reads JSON files with OCR information from an input directory, transforms them according to the index scheme, and sends them to the OpenSearch service for indexing.
- `index_faces.sh` Reads the JSON files with the face information from an input directory, transforms them according to the poster and face index schemes, and sends them to the OpenSearch service for indexing.
- `index_imgfeatures.sh` Reads the JSON files with the image features from an input directory, transforms them according to the index scheme, and sends them to the OpenSearch Service for indexing.
- `index_objects.sh` Reads the JSON files with the data of the recognized objects from an input directory, transforms them according to the index scheme, and sends them to the OpenSearch Service for indexing.

Indexing the metadata of the entire data set of approx. 185,000 posters took 7 minutes in a (non-performance-optimized) development environment. The response times of the metadata and color similarity search are in the range of 30-40ms. There is no noticeable difference to an index with only 500 posters. This shows the scaling behavior of OpenSearch. The proposed index scheme and the implementation of the proposed prototype will be published on Github.

## III. Computer Vision Approaches

The proposed CV approaches to generate additional features are described in detail in the following sections. Qualitative and quantitative analysis is done on subsets of the dataset of 5,000 posters, which have been selected by the experts and are representative for the entire collection. Examples of


Fig. 4. Poster examples.

the posters can be seen in Fig. 4, also showing the variance of the characteristics of the poster collection, reaching from advertising to movie posters, or text-like announcements.

### A. Image Retrieval

Image retrieval allows searching for similar posters compared to an uploaded reference image (query image). Standard Content-Based Image Retrieval (CBIR) systems are summarized in [1], [2].

A standard methodology uses pre-trained Convolutional Neural Networks (CNN) to extract meaningful visual image features. Within this prototype, we have used a ResNet152 which has been pre-trained with the ImageNet dataset [3]. The last layer of the pre-trained model is dropped, and the output of the Average-Pooling-Layer represents the 2,048-dimensional feature vector corresponding to a specific poster. Similar approaches have also been presented by [4], [5]. The feature vector is pre-calculated for the entire poster data set and compared to the feature vector of the query image using the euclidean distance.

A qualitative result is shown in Fig. 5. The first image (left) represents the query image which is followed by a sorted


Fig. 5. Image retrieval example.


Fig. 6. Different font sizes of a poster scaled to 4,096 pixels.

list of retrieved images according to the distance. It can be seen that in the first example (upper row), all retrieved images contain persons with a hat and a subtitle, equivalent to the query image. In the second example, all similar advertising posters are present in the retrieval result. After the second retrieved image, the distance drops since only 2 "Ovomaltine" posters are present in the dataset.

### B. Text Recognition

For text recognition, the open-source OCR engine Tesseract 4[11] was used, which is based on a Long Short-Term Memory (LSTM) architecture and represents a standard for text recognition. Tesseract was an HP prototype and is now developed by Google (2006-2018) [6]. LSTM belongs to the class of Recurrent Neural Networks (RNN). Tesseract is based on an Apache 2.0 license. A general benchmark study with English and Arabic text and different kinds of noise has been presented by [7].

To evaluate the accuracy of Tesseract 4 on the poster data, a subset of 100 posters has been annotated using Aletheia[12], which was developed by the Pattern Recognition and Image Analysis Research Lab (PRImA) at the University of Salford, Manchester. For the GT, the position of the words, as well as the text, were manually annotated. The chosen test set contains different fonts and text sizes. The test set has the following characteristics regarding the fonts:

- 3 posters with predominantly *Fraktur*, which is is a calligraphic hand of the Latin alphabet (see e.g. Fig. 6)
- 15 posters with predominantly Latin script
- 82 posters with predominantly *artistic* text (distorted lettering, lettering with texture, drawn lettering, see Fig. 4)

In addition to the fonts, the resolution of the posters was examined concerning the recognition rate of Tesseract to obtain the best possible result. This is done due to the variance of the font size present in posters, see Fig. 6. If fonts are too large or too small, Tesseract's layout analysis misses the text.

The results are summarized in Table I. It can be seen that for posters with Latin script, a Character Accuracy (CA) of 64.01% and for posters with Fraktur, a CA of 64.88% is achieved with standard settings. The CA of the entire poster data set is below 12% due to the main presence of artistic text (e.g., see Fig. 4 in the middle). The best result on the overall set is achieved with the posters scaled to 4,096 pixels (longer side) which results in a CA of 14.81%. In that case, the main text of posters has a size of 44 pixels to 125 pixels (see Fig. 6). This is also stated by analysis, which shows that the best result is achieved if the height of a capital letter is in the range of approximately 20-50 pixel[13].

TABLE I.
TESSERACT OCR RESULTS.

| **Dataset** | **Character Accuracy** |
|---|---|
| Fraktur | 0.6488 |
| Latin script | 0.6401 |
| Artistic script | $\leq$0.12 |
| **Resolution (entire dataset)** | |
| 1,024 | 0.128 |
| 2,048 | 0.137 |
| 4,096 | 0.1481 |
| 8,192 | 0.1442 |

Thus, for the collection of the Vienna City Library all posters have been re-scaled to 4,096 pixel where tests have shown that a CA of 64% can be reached. Especially posters which contain mainly text can be enriched with the recognized text leading to an advantage regarding search capabilities. The text and its positional information is stored in JSON files and made available to the OpenSearch index.

### C. Face Detection and Retrieval

Face detection and retrieval deal with the detection of faces present on posters. The user can also upload a face query image, and the result is a ranking of all posters with similar faces (retrieval).

A multi-task cascaded CNN is used for face detection and facial landmark alignment proposed by [8]. Zhang et al. report a validation accuracy of 95.4% (O Net) [8]. Based on the detected face, a 512 dimensional feature vector (ArcFace)[14]

[11]https://github.com/tesseract-ocr/tesseract

[12]https://www.primaresearch.org/tools/Aletheia

[13]https://tesseract-ocr.github.io/tessdoc/ImproveQuality.html

[14]https://github.com/deepinsight/insightface/tree/master/recognition/arcface_torch

[9]–[11] is determined for each face, which can be used for retrieval. The cosine distance is used as the distance for the similarity search. All posters are pre-calculated and stored in the OpenSearch index.

Fig. 7 shows a qualitative result of the implemented method [8] in the prototype where all faces and orientations have been correctly detected.


Fig. 7. Face detection and orientation.

Fig. 8 shows a qualitative result of the face retrieval using a cropped face of the Dutch violinist André Rieu. It can be seen that the first 7 results are posters of André Rieu concerts. The confidence score drops for the posters after position 8, which content is unrelated to the search. In the prototype, selecting a poster and visualizing the corresponding face is possible if more people are present within one image.

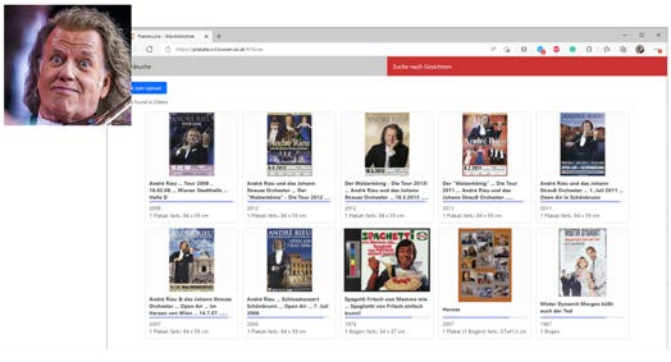

Fig. 8. Result of the face retrieval.

### D. Object Detection

Object detection deals with the detection of instances of defined objects (semantic categories), e.g., humans or clocks, glasses, etc., in images. This is relevant to a search for posters that show particular objects. Zhao et al. [12] present an overview of object detection methodologies. YOLO (You Only Look Once) is a state-of-the-art object detection framework. YOLOv4 is presented in [13]. The prototype of the Vienna City Library uses YOLOv5[15] which is optimized regarding speed. The network is trained on the MS COCO (Common Objects in Context) data set [14], which has about 80 object categories like person, car, chair, book, bottle, etc., and 330K images. YOLOv5 has an AP of 55 on the COCO data set, and a general evaluation of different benchmarks is presented in [15].


Fig. 9. Object detection.

Fig. 9 shows a qualitative result of the YOLOv5 object detection on the poster data set. It can be seen that 4/5 people have been correctly detected. Additionally, 7/10 beer glasses are detected as a cup (2 as wine glasses). The tie of the second person is also correctly detected. The pretzel is misclassified as pizza. The result of the object detection is also encoded as JSON file for the OpenSearch index.

### E. Color Similarity

The color similarity determines the 2 most dominant colors in a poster using k-Means clustering [16]. The colors are weighted according to the number of pixels in relation to the image size. After selecting the color using a color picker (see Fig. 10), the Euclidean distance is determined in the L*a*b* color space. The ranking is done according to the distance value.

For the search, both the recognized primary color and the secondary color are taken into account. A qualitative result is shown in Fig. 10. Additionally, the primary colors are visualized by small circles in the prototype.

[15]https://github.com/ultralytics/yolov5

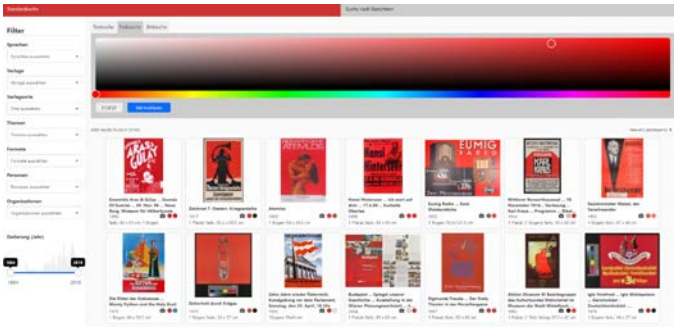

Fig. 10. Color clustering result.

## IV. Conclusion

This paper presents an exploration tool that can be used in libraries, archives, and museums which hold image-based collections. In addition to the traditional metadata search, state-of-the-art CV methods have been used to add visual features as metadata to a search index. In detail, face detection and retrieval, image retrieval, text detection, object detection, and color similarity are presented to allow for extended search capabilities. Experiments with OpenSearch show a *real-time* search behavior on the presented data set. The prototype has been applied to the poster collection of the Vienna City Library but can also be applied to different collections which contain images of objects or text. A prototype is available at https://plakate.cvl.tuwien.ac.at. The quantitative and qualitative assessments show the potential of the added search capabilities for experts and everyday users in a cultural heritage environment.

## Acknowledgment

We would like to thank Julia König and Michael Ingruber of the Vienna City Library for their support.

## References

[1] S. Khawandi, F. Abdallah, and A. Ismail, "A survey on image indexing and retrieval based on content based image," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 2019, pp. 222–225.

[2] S. R. Dubey, "A decade survey of content based image retrieval using deep learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 5, pp. 2687–2704, may 2022.

[3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015. [Online]. Available: http://dx.doi.org/10.1007/s11263-015-0816-y

[4] D. Helm., F. Kleber., and M. Kampel., "Graph-based shot type classification in large historical film archives," in *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP,*, INSTICC. SciTePress, 2022, pp. 991–998.

[5] M. Rashad, I. Afifi, and M. Abdelfatah, "Content-based medical image retrieval based on deep features expansion," in *2022 5th International Conference on Computing and Informatics (ICCI)*, 2022, pp. 331–336.

[6] R. Smith, "An overview of the tesseract ocr engine," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2, 2007, pp. 629–633.

[7] T. Hegghammer, "OCR with tesseract, amazon textract, and google document AI: a benchmarking experiment," *Journal of Computational Social Science*, vol. 5, no. 1, pp. 861–882, nov 2021.

[8] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, Oct 2016.

[9] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.

[10] X. An, J. Deng, J. Guo, Z. Feng, X. Zhu, J. Yang, and T. Liu, "Killing two birds with one stone: Efficient and robust training of face recognition cnns by partial fc," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 4042–4051.

[11] Z. Zhu, G. Huang, J. Deng, Y. Ye, J. Huang, X. Chen, J. Zhu, T. Yang, J. Lu, D. Du, and J. Zhou, "Webface260m: A benchmark unveiling the power of million-scale deep face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 492–10 502.

[12] Z. Zhao, P. Zheng, S. Xu, and X. Wu, "Object detection with deep learning: A review," *CoRR*, vol. abs/1807.05511, 2018. [Online]. Available: http://arxiv.org/abs/1807.05511

[13] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020. [Online]. Available: https://arxiv.org/abs/2004.10934

[14] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2014.

[15] M. Karthi, V. Muthulakshmi, R. Priscilla, P. Praveen, and K. Vanisri, "Evolution of yolo-v5 algorithm for object detection: Automated detection of library books and performace validation of dataset," in *2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, 2021, pp. 1–6.

[16] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 1st ed. Springer, 2007.