ISSN 2464-4617 (print)
ISSN 2464-4625 (DVD)
Computer Science Research Notes
CSRN 3201
WSCG 2022 Proceedings

Semi-automatic Acquisition of Datasets for Retail Recognition

Marco Filax, Tim Gonschorek, Frank Ortmeier
Otto von Guericke University Magdeburg
Universitätsplatz 2
Germany, 39106, Magdeburg, Sachsen-Anhalt
firstname.lastname@ovgu.de

ABSTRACT

The acquisition of datasets is typically a laborious task. It is challenging, especially if the required annotations in every image in the dataset are vast. It is even more challenging if the inter-class variance, the visual difference between two distinct classes, is low. Retail product recognition constitutes an example of both issues. Products are densely packed on shelves, resulting in many objects within an image. Products share visual similarities, which makes them hard to distinguish.
In this work, we propose Annotron, a tool tackling the acquisition problem in this domain. Exploiting dataset structures, such as being organized in consecutive frames, we detect real-world objects through pre-trained detectors and reproject detections to generate candidate traces over time. Further, we aid labelers by computing potential matches of real-world objects and reference images based on their visual similarity: We cluster consecutive detections based on a large set of reference images using embeddings acquired from pre-trained networks.
Using the proposed tool reduces manual efforts drastically by diminishing the time spent on repetitive, error-prone tasks. We evaluate Annotron in the retail recognition domain. The domain is commonly considered fine-grained, which means that instance-level annotations are costly due to the described problems. We refine the given dataset, surpass the number of previously found stock-keeping units, and label over 446.500 individual bounding boxes.

Keywords

Dataset Acquisition, Pattern Recognition, Smart Assistance, Retail Product Recognition

1 INTRODUCTION

Collecting datasets for supervised learning tasks is a laborious but mandatory task. It typically requires human data labelers to judge vast amounts of data points such that a model can learn to make correct decisions. In the context of computer vision, we typically let labelers annotate images with bounding boxes and classes, i.e., visual concepts represented by regions of pixels. Further, additional techniques might be required that help to increase the overall accuracy, such as labeler consensus or label auditing. These techniques increase the overall time investment to design a particular solution for new supervised learning tasks

The acquisition of datasets is extremely costly in the fine-grained domain. It requires solid precision to determine the correct visual concept of a particular image region. The problem of retail product recognition gives a perfect example of a fine-grained domain. Figure 1 depicts an exemplary image of this domain. It visualizes two issues that arise in this fine-grained domain: First, images of retail products on shelves comprise crowded scenes. Similar-looking products are densely stacked across the complete frame. Annotating a single frame is a tedious task due to the sheer number of ob-

Figure 1: Retail product recognition is a supervised fine-grained visual learning task. Crowded scenes and low visual inter-class variance require significant manual annotation efforts. We propose *Annotron*, a tool that lowers these hurdles in fine-grained domains.

jects. Second, some products can only be distinguished through small visual cues. Both reference images illustrate this problem: although they are entirely different objects, they share significant visual similarities. If meta-data of the reference set is available: labelers have to annotate the visual concepts based on an exhausting full-text search over the product's name. If meta-data is not available: the situation is even worse. Labelers would have to select the correct match based on visual cues. Semi-automatic approaches could tackle both problems.

[5] surveyed related works in the domain of (semi-)automated image annotation. We focus on more recent works since the authors already gave a broad review of works proposed until 2017. More recently, works have been proposed that use the labeler in a loop in conjunction with trained networks [1], [2], [14], [16], [26]. Here, the complete dataset is typically split into two sets. The first is manually annotated and used to train networks which then predict labels for the latter. Other works extend this approach by fuzzing the dataset split, i.e., by predicting the best video frames to annotate manually [15]. [17] proposed to use properties of existing detectors, which tend to predict multiple different-sized bounding boxes for a single object. The user has then to choose the correct one iteratively. [21] proposed to use GrabCut [19], which is designed to segment an object based on user input. Using additional meta-knowledge, i.e., the depth stream of a video stream, [21] proposed to track the segmented object in 3D space and reproject bounding boxes onto the image planes. [7] used meta-knowledge of the environment to propose a semi-automatic annotation tool tailored for retail environments. The authors used a SLAM approach to annotate objects in 3D space and reproject their annotation onto consecutive video frames. Related works focus on general (semi-)automated image annotation to our knowledge. Our approach does not need any additional knowledge about the environment and is tailored to crowded scenes, in which mainly the manual annotation of individual objects is time-consuming and error-prone.

In this work, we propose a semi-automatic image annotation system called *Annotron*, with the scope of allowing a fast and continuous annotation workflow. We evaluate the proposed approach in the fine-grained domain of retail recognition, in which acquiring instance-level annotations is considered costly. We propose exploiting underlying structures of datasets by reprojecting found bounding boxes to consecutive frames. This approach yields a candidate stream of a particular object over time and extracts multiple views of the same object. We extract embeddings, a lower-dimensional representation of the visual content, of every image patch in the candidate stream. With these, we form groups of similar-looking visual concepts, e.g., retail products that share similar looks or real-world images that look similar to a particular reference image. We gather the set of nearest neighbors for every candidate stream. A labeler finally identifies the correct reference image to acquire the ground-truth annotation.

The proposed approach efficiently lowers the need for manual assistance. It enables labelers to annotate different views of the same stock-keeping unit simultaneously. Further, it reduces the search space dramatically from which the labeler has to choose the correct reference concept. We achieve this by using a mutual embedding space of candidates and reference images. We demonstrate the ease of use throughout the paper. Our experiment shows that the proposed approach combined with lower annotation efforts **extends** a given database. We found more visual concepts using fewer annotation efforts as proposed in the original work.

The paper is structured as follows: Section 2 explains the proposed approach in detail. We describe the required preprocessing step in-depth and elaborate on the resulting tool support that differentiates manual annotation tasks. Afterward, we evaluate the proposed approach with an already given dataset in Section 3. We report on the specific choices that arise in the preprocessing step and demonstrate that the resulting new dataset extends the previously given dataset. Finally, we conclude our work.

2 ANNOTRON

We propose a semi-automatic labeling tool to ease the hurdle of acquiring few-shot datasets focused on fine-grained recognition problems. We follow a two-stage approach: First, we preprocess the original data to identify similar-looking regions across multiple frames as described in Section 2.1. This is done in a fully automated manner. Second, we label these region tracks manually, which we call candidate traces, using different labeling strategies. This allows us to annotate vast amounts of previously tracked candidate traces efficiently. Further, we automatically cluster similar-looking traces to reduce the manual search time invested. We describe the manual tasks in Section 2.2.

2.1 Preprocessing

The first stage of the proposed approach is a fully-automated preprocessing process that aims at generating an intermediate data structure. The core idea is to automatically extract high-level representations of subsequent image regions and determine similar visual concepts. We group and present these in an ordered manner to the labeler (cf. Section 2.2).

The basis for the proposed preprocessing step is the assumption that a dataset is organized in a consecutive manner, i.e., in video streams. Further, we assume

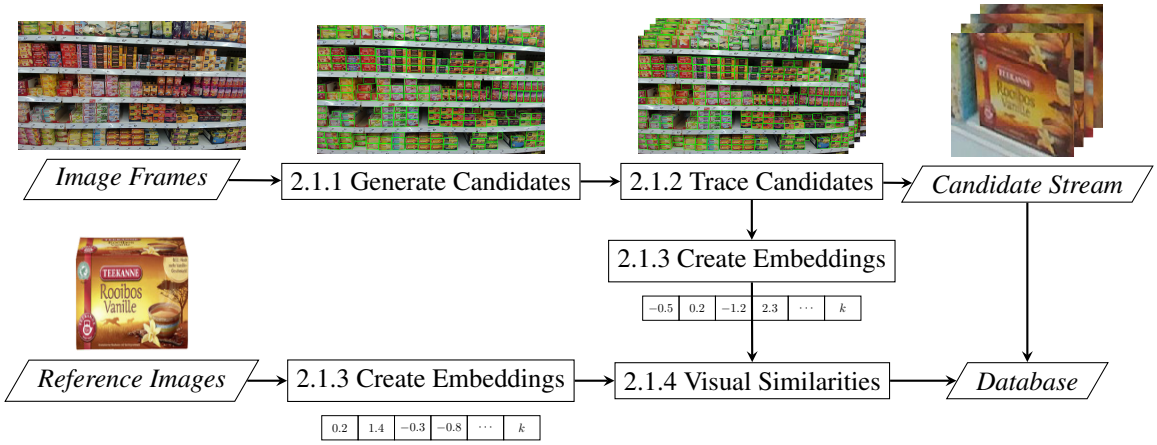


Figure 2: Flowchart of the proposed preprocessing module. We trace candidates across consecutive frames and use embeddings to find similar-looking image patches. We comprise these findings in a database for fast user access.

that at least a single reference image of all relevant visual concepts is available. Figure 2 depicts the proposed preprocessing module. Generally, we propose reprojecting given candidates, e.g., acquired with a generic detection module, onto consecutive frames (cf. Section 2.1.1). Using the resulting candidate streams (cf. Section 2.1.2), we encode these and the reference images using an embedding model (cf. Section 2.1.3). Finally, we extract the nearest neighbors (cf. Section 2.1.4) of every candidate stream and comprise the results in a special data structure.

2.1.1 Generate Candidates

The first step in the proposed preprocessing module is generating possible object candidates on every real-world image. Generally, we assume that an image holds multiple objects of interest. We propose acquiring possible candidates, i.e., arbitrary shapes or volumes in the input space, using a general-purpose detector, if possible. The detector does not need to predict the correct class or concept of a particular object of interest.

In this work, we focus on images as input. We consider a bounding box the most common form of shape that needs to be detected for every frame. Consequently, we use the detector to predict bounding boxes for as many objects as possible. We emphasize the possibility of tailoring the proposed preprocessing module as needed, i.e., replacing the detector with a shape predictor. This might be necessary if the objects of interest are to be described in a different form.

2.1.2 Trace Candidates

We aim at lowering the amount of manual labeling activities. Therefore, the core idea is to trace object candidates - regions within every frame that depict similar visual concepts - across multiple frames. We use the well-known overlap measure intersection over union. Given two shapes $A, A' \in R^2$, the IoU is defined as

$$IoU = \frac{|A \cap A'|}{|A \cup A'|} \tag{1}$$

We iteratively trace two shapes, i.e., bounding boxes, A and A' of two consecutive frames. We select the overlapping bounding boxes with a watershed algorithm while maximizing the IoU. Thereby, we generate candidate streams - traces of real-world objects over time - by iterating over all images of a video to gather different views of the same object. We empirically found that this approach seems reasonable precise for video streams from the retail domain. However, we maintain the possibility to slice a candidate stream manually through the labeler.

2.1.3 Create Embeddings

Next, we aim at clustering found candidate streams to find visual cues of similar objects shown in the image regions. To do so, we create a vectorized representation of the visual content of every image in the candidate stream. We use deep neuronal nets to create embeddings, a lower-dimensional representation of the visual content. This is a common approach for many specialized fields, such as face recognition [6], [20], [27] or person recognition [3], [13], [22], that typically deal with few-shot learning problems.

[4], [24], [25] have shown that generic classification networks can be used for a few shot recognition tasks, with and without any additional training. The authors proposed removing the network's classification head and using the penultimate output as embeddings. For the domain of retail recognition, there also exist specialized neuronal networks [8], [9], [23] designed to produce tailored embeddings. We use a specialized network wherever pre-trained weights tailored to the domain are available and use generic embedders otherwise.

2.1.4 Visual Similarities

Generally, the proposed approach uses the fact that embeddings of similar concepts are mapped to close regions in the embedding space. We exploit the observation by mapping real-world traces and reference images into a mutual embedding space. We then select the nearest reference images for every sample within a candidate stream. Further, we gather the nearest candidate streams of each candidate stream. Here we use the center point of the embeddings, i.e., the mean embedding of the complete candidate stream, because we assume that a candidate stream depicts the same visual concept over time.

We gather the top set of nearest neighbors in both cases, assuming that in the top nearest neighbors most likely is the true visual concept. Given the underlying problem of retail recognition with its fine-grained nature, we found that retrieving the five nearest reference images whereas 50 nearest candidate streams of every candidate stream yield feasible results. We gather all nearest neighbors in a data structure, i.e., using identifies, for fast and easy access during the labeling procedure.

2.2 Labeling Procedure

The second step of the proposed process is the actual labeling, i.e., linking real-world objects with visual concepts. Labeling vast amounts of data is a monotone, time-consuming, and error-prone task. To overcome some of the hurdles induced through the necessity of training with labeled data, we proposed a preprocessing step in the previous Section 2.1. The core idea is to use the previously computed similar-looking candidate streams with their most similar-looking reference images and present this information in an ordered manner to the user. We thereby focus on annotating complete candidate streams with minimal user interaction.

The ability to operate on candidate streams rather than on individual image regions yields different benefits during the labeling process. First, it produces a particular relevance scheme for all candidate streams, which induces an order. Second, it enables us to cluster candidate streams through computing similarity measures. This allows us to identify similar visual concepts from different directions, i.e., previously labeled candidate streams that share joint visual embeddings are more likely to depict the same visual concept.

Ordering the labeling procedure yields soft benefits during processing, such as faster feedback loops that increase morale. We inherit a better ability to monitor the labeling activities because the number of manual interactions is greatly reduced. It is becoming easier to reach achievements.

Using clustered candidate streams further allows us to distinguish two different types of labeling tasks: *Identifying new visual concepts* in the data that have not been linked to visual concepts and *increasing the number of observations* of already seen concepts. In the following, we present both tasks in detail and elaborate on the design choices for each task.

2.2.1 Task 1: Identify Visual Concepts

When we identify new concepts in the data, we link previously unseen reference concepts to candidate streams. Without *Annotron*, we would have to detect new concepts based on the human eye and a full-text search over the reference names if the labeler can identify them. This is error-prone due to the fine-grained nature of the problem. Using *Annotron* provides better tool support to the user while identifying new concepts by presenting similar-looking reference concepts. Presenting visually similar reference and candidate images removes the burden of full-text search activities and, therefore, increases the overall labeling speed of a single labeler.

We propose specialized tool support for this particular task to rank the previously computed candidate streams according to two different metrics, which are defined as follows:

Tracking Stability: We weigh the candidate streams with the longest stable tracking to be the most relevant. Given a candidate stream $c_i \in C$, we define the metric m_t as $m_t(c_i) = -|c_i|$. This metric focuses on fast annotations that quickly maximize the total number of observed examples per concept in the dataset.

Embedding Stability: We sort the candidate streams according to their visual stability. We assume that a candidate stream depicting a particular concept available in the set of reference images will lead to stable nearest neighbors in the embedding space of reference images. Given a candidate stream c_i, with multiple nearest reference concepts N_{c_i} in the embedding space, we define the metric m_e as $m_e(c_i) = \frac{|\{N_{c_i}\}|}{|c_i|}$.

2.2.2 Task 2: Increase Observations of Concepts

The second annotation task increases the total number of annotations of a particular concept within the dataset. The user manually identifies the previously found reference concepts on other video stream regions in the dataset. Without *Annotron*, we would again have to identify a reference concept of a particular candidate stream based on the human eye and full-text search. This task would not differ from the previous and might induce errors in the annotation process.

Using *Annotron* allows us to present similar-looking reference images when viewing a candidate stream. This dramatically reduces the number of full-text queries a labeler has to issue during the annotation

procedure. It also inherits a reduced risk of annotation errors. Further, when the labeler selects a reference concept, that is already assigned to a stream, we display similar-looking candidate streams. This decreases the effort of identifying similar-looking ones in other streams. We distinguish two different types of similarity relations:

Similarity to other Candidate Streams: To describe the visual distance of two candidate streams, we use the center of the hyperspheres in the embedding spaces formed with all embeddings in a stream. To do so, we use the mean embeddings μ of a candidates stream $c_i \in C$ which are defined, following [11], as $\mu_i = \frac{1}{|c_i|}\sum_j^{|c_i|} z_j$, whereas z_j is the embedding of an image in the candidate stream. We measure the distance of candidate streams as $distance(c_i, c_j) = ||\mu_i - \mu_j||_2^2$.

Similarity to Reference Concepts: Further, we propose to measure the distance of a candidate stream and reference images. We measure the distance of a candidate stream and a reference embedding z_r similarly, $distance(c_i, z_r) = ||\mu_i - z_r||_2^2$. This allows us to present candidate streams visually similar to reference concepts during the labeling process.

Other works do typically not differentiate these two tasks. We, however, assume that modern visual neuronal nets produce embeddings powerful enough to distinguish the visual content of image patches to some extend. This is justified by observing that these approaches are used in broad scopes, i.e., face, human, or character recognition and anomaly detection. Using visual similarity to assist the user dramatically reduces the mental efforts of an annotator while labeling image patches. We found the ability to distinguish different labeling tasks to be the most dominant benefit. It enables us to differentiate the tool support necessary for every individual step to increase the overall efficiency of the manual input.

We implemented the proposed approach in a tool called *Annotron*. An example is shown in Figure 3. We evaluate the proposed approach using a case study from the retail recognition domain.

3 CASE-STUDY

In this section, we present the implementation of the proposed tool called *Annotron*. We evaluate the proposed tool in the fine-grained domain of retail product recognition. First, we describe an existing dataset that will serve as a basis for our work in Section 3.1. Second, we elaborate on the concrete implementation of the automatic preprocessing module. Third, we give insights on the manual work in Section 3.3. Finally, we evaluate the outcome based on the resulting dataset and compare it to the original dataset.

3.1 A Retail Recognition Dataset

We refine an already existing dataset to evaluate the proposed approach. We chose a fine-grained dataset from the domain of retail product recognition. In our opinion, fine-grained datasets cover a very challenging acquisition problem. Thus, we chose to refine a semi-automatic generated large-scale dataset taken from [7] to illustrate the efficiency of the proposed approach.

The original dataset contains over 23.000 different reference images and offers 41.000 frames from more than 20 video sequences. They were gathered using a webcam mounted to a hololens. The authors used additional meta-knowledge of the environment to label the data semi-automatically. The authors found 871 different stock-keeping units in the database, collected with four labelers.

Unfortunately, the dataset does comprise some inaccuracies. Bounding boxes were tracked over time using the camera's position calculated with the internal measurements of the hololens. This leads to some irregular annotations, possibly due to synchronization issues. Further, the authors deployed standard tool support to identify visual concepts, such as a full-text search using the products' names.

We assume that there is some headroom for improvements regarding the complete localization of available reference concepts, i.e., products. We find it challenging to determine the fine-grained visual differences of retail products because products typically share significant visual aspects and the enormous reference concept space. This makes the Magdeburg Groceries Dataset [7] the perfect basis for our experiments.

3.2 Automatic Preprocessing

This section describes our implementation of the proposed automatic preprocessing module. Further, we present detailed information on the parametrization.

3.2.1 Generate Candidates

Following the proposed approach, we generated bounding boxes for every frame as proposed in Section 2.1.1. We used a pre-trained network proposed in [18]. It was trained on the SKU-110k dataset [10]. The dataset is taken from a similar domain - retail product *detection*. Thus, it can acquire the location of retail products on shelves. We chose to use the network with pre-trained weights due to the pure availability of a pre-trained detector. However, it cannot recognize the classes of products in any sense. Thus, it serves as a good basis to predict bounding boxes in the retail domain.

3.2.2 Trace Candidates

The previously predicted bounding boxes are mapped from one frame to the next. In this step, the goal is

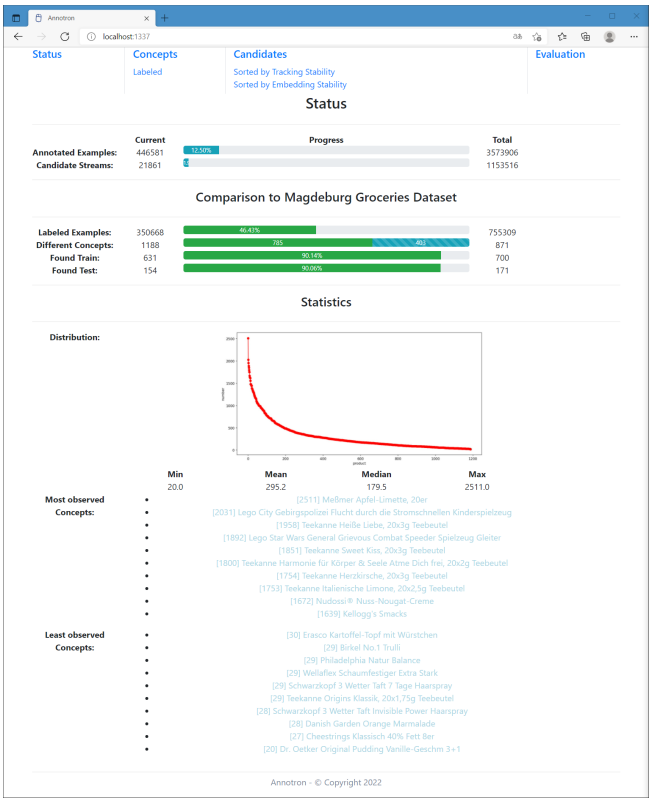


Figure 3: We implemented *Annotron* as a web service for easy access and minimal system requirements. The status page depicts various statistics of the current dataset.

to interconnect the locations across multiple frames. As proposed in Section 2.1.2, we greedily select the bounding boxes by calculating the *IoU* of consecutive frames. Thereby we maximized the overlap based on iteratively selecting bounding boxes that overlap with a watershed algorithm. We accepted consecutive traces up to a threshold of 0.5. The consecutive trace of overlapping bounding boxes, i.e., candidate traces, is fed into the embedding network to describe the visual content.

3.2.3 Create Embeddings

We describe the visual content of candidate traces using embeddings to assist the user during the error-prone manual work. We proposed using an existing network to generate embeddings of visual inputs in Section 2.1.3. We again use an already-trained network [8] from the retail domain. The network architecture is based on the well-known resnet-50 architecture [12]. The classification head was removed, and a 128-dimensional embedding head was attached. We embed the visual content of every reference concept, and every candidate of all traces scaled to 128x128 pixel patches. We use the euclidean distance metric to compare these embeddings.

3.2.4 Visual Similarities

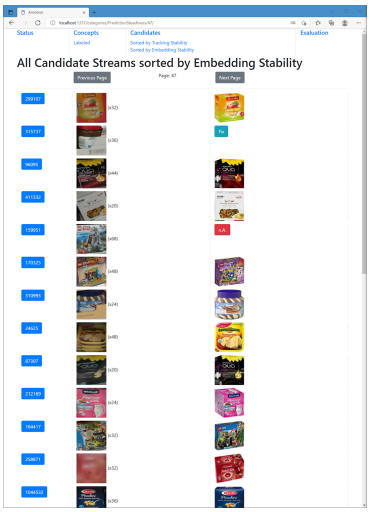
The last preparation step is the identification of visual similarities. As proposed in Section 2.2, these embedded visual representations are used to assist the labeling procedure later. As shown there, we are using the k-nearest neighbors of reference and candidate images as well as the k-nearest neighbors of the mean of candidate images. We use an approximated variant[1] since retrieving k=50 nearest neighbors can be a resource-intensive task. We precompute the nearest neighbors and save their identifiers in a database to allow a smooth user experience.

3.3 Manual Annotation

Finally, it is the task of the labeler to identify visual concepts and annotate them manually. Figure 3 depicts the status page of the proposed tool. It allows the labeler to monitor the labeling progress itself and the distribution of tagged candidate streams.

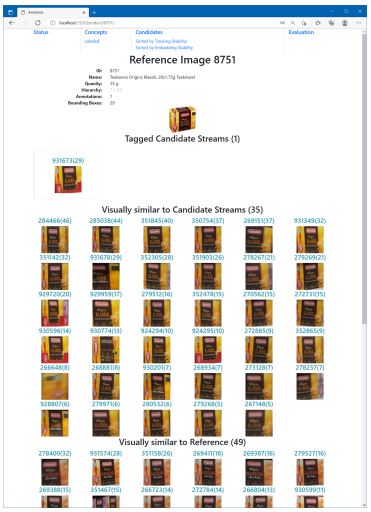
Further, it displays a comparison to the original Magdeburg Grocery dataset [7]. We identified 1188 visual concepts, i.e., retail products. The original work found 871 visual concepts in the data. We assume that the difference is mainly because the original work used standard tool support to identify products, such as a full-text search over product names. We instead used the visual similarities of image patches. Thus, we conclude that the larger number of found products must be due to the better tool support.

[1] github.com/spotify/annoy



(a) Identify Visual Concepts. *Annotron* comprises special tool support that structures the annotation workflow. Tracked candidate streams are ordered based on their embedding stability.

(b) Annotation View. *Annotron* depicts all samples of a candidate stream and the nearest neighbors based on visual similarity.



(c) Increase Observations. The user can quickly identify more visually similar candidate streams using already tagged streams or the reference image of the visual concept 8751.

Figure 4: Different stages of the annotation process depicted in *Annotron*. *Annotron* provides specialized tool support for the different stages in the labeling process.

Figure 4 depicts the proposed tool support at a glance. In this work, we describe the three major views of *Annotron*. We cover the particular views of *Annotron* in the following.

3.3.1 *Identify Visual Concepts*

Figure 4a depicts the user interface of *Annotron* during the identification phase. Here, the candidate streams are ordered by embedding stability. A single click redirects the user to Figure 4b.

On the left side of Figure 4b all individual image patches of the candidate stream are shown. The right side of *Annotron* displays a distinct list of the top-5 nearest neighbors of every image patch in the candidate stream ordered by their number of votes. A single click links the visual reference concept to the candidate stream. *Annotron* redirects the user to the next candidate stream (cf. Figure 4b) without additional interaction.

3.3.2 *Increase Observations of Concepts*

Figure 4c depicts the user interface of *Annotron* designed explicitly for the phase of increasing the observations of the particular visual concept. The GUI is organized similarly to a classical website. At the top, all information of the visual concept is presented. After that, already annotated candidate streams are depicted. Then, candidate streams that are similar to already tagged streams are depicted. With a single click, the user can link the displayed candidate streams to the visual concept. If the user hovers over a candidate stream, we virtually scroll through it, enabling him to identify invalid patches easily. Finally, the nearest neighbors of the visual concept in the embedding space are depicted at the bottom of the page. In the particular example, we can see the challenges of fine-grained recognition problems: Close neighbors in the candidate stream space depict visually similar classes to already tagged candidate streams, but the nearest neighbors of the reference image depict purely invalid samples that share large visual similarities. It is the duty of the labeler to identify only valid matches and link real-world and reference images accordingly.

3.4 Refined Dataset

This section covers the final result of our labeling activity. We detected 3.573.906 images patches in the complete set of image frames. We further managed to track 1.153.516 candidate streams. We annotated 446.481 image patches of 1.188 different retail products. Thereby, we manually annotated 21.861 candidate streams, meaning that every stream we manually labeled consists of 20 tracked bounding boxes on average. That underlines the greedy strategy we employed through the labeling phase.

As shown in the *Annotron* tool in Figure 3, we managed to identify over 90% of the previously found visual concepts. This is especially remarkable since we annotated the retail products with lower manpower as in the original dataset. We labeled the data with a single labeler,

while four labelers were needed in the original work. Further, we identified over 400 new visual concepts that were already presented in that dataset and not labeled in the original variant. We conclude that this increase of found retail products has to originate in the better tool support proposed in this work.

4 CONCLUSION

In this article, we proposed an semi-automatic image annotation tool called *Annotron*, which aims at lowering the hurdle of acquiring new datasets in fine-grained domains. We exploit the structure in datasets, such as in the domain of retail product recognition, in which products are densely packed on shelves, to achieve a fast and continuous annotation workflow by reprojecting bounding boxes of consecutive image frames. This is also helpful in other datasets if they are similarly organized in video frames.

Further, we exploit the capabilities of modern neuronal networks by projecting the visual contents of reference and real-world images into a mutual embedding space. This enables us to extract similar-looking objects and determine possible matches, i.e., depicting the same visual concept, from both input spaces. We implement an intuitive interface that presents possible matches to a labeler to acquire ground truth annotations of objects tracked over time.

We demonstrated the applicability of *Annotron* in the fine-grained domain of retail recognition. We refined an existing database from the literature and extended the total number of found stock-keeping units with lesser manual effort. We showed that the proposed approach efficiently lowered the need for manual assistance during the labeling procedure.

However, we only focused on a single database and heavily relied on the fact that it consists of videos. If the images to be tagged are not consecutively ordered, we cannot track patches across multiple frames, which prevents us from finding candidate samples and ultimately increases the required annotation efforts. The effort gains originated in the proposed idea of using encoded image representations to acquire possible matches remain in effect. We plan to address this validity flaw by extending another database in the future.

REFERENCES

[1] B. Adhikari and H. Huttunen, "Iterative bounding box annotation for object detection," *Proc. - Int. Conf. Pattern Recognit.*, pp. 4040–4046, 2020. arXiv: 2007.00961.

[2] B. Adhikari, J. Peltomäki, J. Puura, and H. Huttunen, "Faster Bounding Box Annotation for Object Detection in Indoor Scenes," *Proc. - Eur. Work. Vis. Inf. Process. EUVIP*, 2019. arXiv: 1807.03142.

[3] X. Bai, M. Yang, T. Huang, Z. Dou, R. Yu, and Y. Xu, "Deep-Person: Learning discriminative deep features for person Re-Identification," *Pattern Recognit.*, vol. 98, 2020. arXiv: 1711.10658.

[4] A. Bendale and T. E. Boult, "Towards Open Set Deep Networks," in *CVPR*, vol. 2016-Decem, IEEE, Jun. 2016, pp. 1563–1572. arXiv: arXiv:1511.06233v1.

[5] Q. Cheng, Q. Zhang, P. Fu, C. Tu, and S. Li, "A survey and analysis on automatic image annotation," *Pattern Recognit.*, vol. 79, no. November, pp. 242–259, 2018.

[6] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," 2018. arXiv: 1801.07698.

[7] M. Filax, T. Gonschorek, and F. Ortmeier, "Data for Image Recognition Tasks: An Efficient Tool for Fine-Grained Annotations," in *ICPRAM*, SciTePress, 2019, pp. 900–907.

[8] M. Filax, T. Gonschorek, and F. Ortmeier, "Grocery Recognition in the Wild: A New Mining Strategy for Metric Learning," in *VISIGRAPP 2021 - Proc. 16th Int. Jt. Conf. Comput. Vision, Imaging Comput. Graph. Theory Appl.*, vol. 4, SciTePress, 2021, pp. 498–505.

[9] M. Filax and F. Ortmeier, "On the influence of viewpoint change for metric learning," in *Proc. MVA 2021 - 17th Int. Conf. Mach. Vis. Appl.*, Institute of Electrical and Electronics Engineers Inc., Jul. 2021.

[10] E. Goldman, R. Herzig, A. Eisenschtat, J. Goldberger, and T. Hassner, "Precise Detection in Densely Packed Scenes," in *2019 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, IEEE, Jun. 2019, pp. 5222–5231. arXiv: 1904.00853v3.

[11] M. Hassen and P. K. Chan, "Learning a neural-network-based representation for open set recognition," in *Proc. 2020 SIAM Int. Conf. Data Mining, SDM 2020*, Society for Industrial and Applied Mathematics Publications, 2020, pp. 154–162. arXiv: 1802.04365.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *CVPR*, IEEE, 2016, pp. 770–778. arXiv: 1512.03385.

[13] A. Hermans, L. Beyer, and B. Leibe, "In Defense of the Triplet Loss for Person Re-Identification," *arXiv:1703.07737*, 2017. arXiv: 1703.07737.

[14] K. G. Ince, A. Koksal, A. Fazla, and A. A. Alatan, "Semi-Automatic Annotation for Visual Object Tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2021-Octob, 2021, pp. 1233–1239.

[15] A. Kuznetsova, A. Talati, Y. Luo, K. Simmons, and V. Ferrari, "Efficient video annotation with visual interpolation and frame selection guidance," in *Proc. - 2021 IEEE Winter Conf. Appl. Comput. Vision, WACV 2021*, 2021, pp. 3069–3078. arXiv: 2012.12554.

[16] T. N. Le, S. Akihiro, S. Ono, and H. Kawasaki, "Toward interactive self-annotation for video object bounding box: Recurrent self-learning and hierarchical annotation based framework," in *Proc. - 2020 IEEE Winter Conf. Appl. Comput. Vision, WACV 2020*, 2020, pp. 3220–3229.

[17] D. P. Papadopoulos, J. R. Uijlings, F. Keller, and V. Ferrari, "We Don't need no bounding-boxes: Training object class detectors using only human verification," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, no. 1, pp. 854–863, 2016. arXiv: 1602.08405.

[18] T. Rong, Y. Zhu, H. Cai, and Y. Xiong, "A Solution to Product detection in Densely Packed Scenes," 2020. arXiv: 2007.11946.

[19] C. Rother, V. Kolmogorov, and A. Blake, ""GrabCut" - Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[20] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *CVPR*, IEEE, 2015, pp. 815–823. arXiv: 1503.03832v3.

[21] D. Stumpf, S. Krauß, G. Reis, O. Wasenmüller, and D. Stricker, "SALT: A semi-automatic labeling tool for RGB-D video sequences," in *VISIGRAPP 2021 - Proc. 16th Int. Jt. Conf. Comput. Vision, Imaging Comput. Graph. Theory Appl.*, vol. 4, 2021, pp. 595–603. arXiv: 2102.10820.

[22] X. Sun and L. Zheng, "Dissecting Person Re-Identification From the Viewpoint of Viewpoint," in *CVPR*, IEEE, 2019, pp. 608–617. arXiv: 1812.02162.

[23] A. Tonioni and L. Di Stefano, "Domain invariant hierarchical embedding for grocery products recognition," *Comput. Vis. Image Underst.*, vol. 182, pp. 81–92, 2019. arXiv: 1902.00760.

[24] R. Vemulapalli and A. Agarwala, "A compact embedding for facial expression similarity," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 5676–5685, 2019. arXiv: 1811.11283.

[25] N. Vo and J. Hays, "Generalization in metric learning: Should the embedding layer be embedding layer?" *Proc. - 2019 IEEE Winter Conf. Appl. Comput. Vision, WACV 2019*, pp. 589–598, 2019. arXiv: 1803.03310.

[26] P. Voigtlaender, L. Luo, C. Yuan, Y. Jiang, and B. Leibe, "Reducing the annotation effort for video object segmentation datasets," in *Proc. - 2021 IEEE Winter Conf. Appl. Comput. Vision, WACV 2021*, 2021, pp. 3059–3068. arXiv: 2011.01142.

[27] M. Wang and D. Weihong, "Deep Face Recognition: A Survey," in *Proc. - 31st Conf. Graph. Patterns Images, SIBGRAPI 2018*, 2019, pp. 471–478. arXiv: 1804.06655v8.