UNIVERSITÉ GRENOBLE ALPES – ENSIMAG

MASTER OF SCIENCE IN INDUSTRIAL AND APPLIED MATHEMATICS

MASTER THESIS

Segmentation of compound figures from biomedical literature

Author:
Gaétan LEPAGE
Tutor:
Prof. Eric Gaussier

Supervisor:
Prof. Henning MÜLLER
Co-Supervisor:
Prof. Manfredo ATZORI

Project realised within MedGIFT research team
from Haute École spécialisée de Suisse occidentale

Haute Ecole Spécialisée
de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences and Arts
Western Switzerland

April – August 2020

Abstract of thesis entitled

Segmentation of compound figures from biomedical literature

Submitted by

Gaétan LEPAGE

for the degree of Master of Science

at Université Grenoble Alpes

in August, 2020

Compound figures consist in visual illustrations composed of several subparts. Scientific literature, and especially medical publications carry a great amount of such figures. As automatic extraction, classification and image processing methods are more and more used on huge amount of data, compound figures need to be segmented to be processed as individual images by more generic algorithms. The absence of formatting rules to organize such figures makes their separation a challenging task. Multi panel figures separation includes various sub tasks that involve dealing with multimodal data. Most of them have been tackled in previous works. The objective of this thesis is to conglomerate efficient techniques to build an effective pipeline for compound figure separation. The developed algorithms include state of the art solutions to solve each of the involved subtasks. A wide overview of previous works is given to attest of the latest developments in this domain. Modern deep learning architectures have been employed to address the visual panel segmentation problem. Some innovative ways of merging sub results are enhancing the overall performance. In addition to handling image data, this solution deals with the caption of the figures. The performance of our proposed methods were evaluated in various settings. Different data sets where used (such as the PubMed Central database and ImageCLEF). Overall, satisfying results are achieved and demonstrate the efficiency of this approach. This work is part of the European funded project ExaMode as a standalone deliverable. The developed open source program aims at offering a strong basis for further developments in complete compound figure separation. Several unexplored solutions are proposed and are expected to be tested to even more enhance the quality of this solution.

Acknowledgments

I would like to thank my supervisors, Prof. Henning MÜLLER and Prof. Manfredo ATZORI for their support and guidance. The COVID-19 crisis provided a challenging context for this Master's project. The entire work was performed remotely. Even though this situation was not ideal and I would have truly appreciated meeting the team members in person, my internship went really well. Both my supervisors have shown themselves responsive to my interrogations, doubts and remarks. A sincere atmosphere of collaboration took place instantly. I have been greatly involved in the daily life of the research team. This internship has been a scientific journey and a learning process.

I would also like to thank the other members of the MedGIFT team I worked with. Niccolò Marini and Sebastian Otálora shared their experience of Phd students and gave me useful advice as well as feedback on my work. During my first attempts at scientific writing, I received sensible recommandations. The team took the time of diving into my technical concerns and helped me design efficient solutions.

Gaétan LEPAGE
Université Grenoble Alpes - Ensimag
August 24, 2020

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 MedGIFT and ExaMode

MedGIFT MedGIFT[1] is a research team from Haute École spécialisée de Suisse occidentale (HES-SO)[2]. The project originally started in 2002 at the medical faculty of the University of Geneva[3]. The focus of the team is to develop state of the art tools to extract information from complex medical data sources. MedGIFT is implied in several projects such as ExaMode, ImageCLEF, MeganePro, GAMBLY, etc (see http://medgift.hevs.ch/wordpress/projects/).
This Master's project was supervised by MedGIFT team leader, Prof. Henning MÜLLER and senior researcher Prof. Manfredo ATZORI.

ExaMode: Extreme-scale Analytics via Multimodal Ontology Discovery & Enhancement The ExaMode[4] project is a research initiative funded by the European Union through the Horizon 2020 framework. As data is coming more and more ubiquitous, several challenges arise. The various sources of raw information are today a key for developing new products and technological advances. Their extraction and exploitation are difficult because of their heterogeneity.

Healthcare provides enormous amounts of data that are expected to reach 2000 exabytes ($2 \times 10^{21} B$) in 2020. The recent advances in machine learning and especially in deep learning provide efficient tools to exploit massive data sets. However, the relevance of those methods highly depends on the quantity and quality of cleanly annotated data. Supervised models have demonstrated impressive results on well-defined tasks for which labeled data has been made available but are more challenging to apply to heterogeneous forms of data.

ExaMode is addressing three objectives [12]:

- Weakly-supervised knowledge discovery for exascale medical data.

[1] http://medgift.hevs.ch
[2] https://www.hevs.ch/
[3] https://www.unige.ch/medecine/
[4] https://www.examode.eu/

- Develop extreme scale analytic tools for heterogeneous exascale multimodal and multimedia data.
- Healthcare & industry decision-making adoption of extreme-scale analysis and prediction tools.

Hence, ExaMode is trying to assess the potential of weakly supervised approaches for dealing with exascale heterogeneous data sets. Several teams are collaborating on those questions. HES-SO stands as the project coordinator. Both academic (University of Padua, Radboud University Medical Center[5], etc.) and industrial (MicroscopeIT[6], SURFsara[7], etc.) actors are participating in the ExaMode project. This thesis deals with a problem of medical information processing and is directly engaged with the ExaMode project (as deliverable 3.2[8]).

Figure 1.1: ExaMode project logo.

1.2 Compound image separation

1.2.1 Background and motivations

Figures represent a fundamental part of scientific literature. They allow humans to better understand the content described in the text of articles and books, thus representing a valuable source of knowledge.

Developing methods that can help to extract knowledge from the figures and text included in scientific articles and books is a problem that is currently unsolved [39] and that is part of the ExaMode project objectives. There are many challenges that make it difficult to extract knowledge from scientific literature. Among those, compound figures separation is one of the biggest and still less studied ones. This thesis clarifies the problem of compound figures separation. It describes what has been done in literature to face it as well as a procedure to fully approach it and evaluates it on concrete data.

Compound figures can be defined as images that include several sub-figures (panels), eventually identified by panel labels (that can be letters, digits, roman numerals, or combinations of them). Figure 1.2 shows a labeled compound figure composed of four panels. Compound figures are usually associated with a textual description (the caption), that in most cases refers to each panel via the labels.

[5] https://www.radboudumc.nl/research
[6] https://www.microscopeit.com/
[7] https://www.surf.nl/en/research-ict
[8] https://www.examode.eu/deliverables/

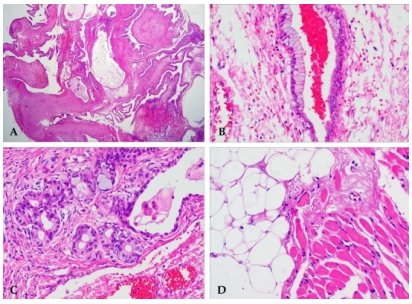


Figure 1.2: Example of a compound figure of four panels with alphabetical labels.

The scientific literature stores a very large amount of available medical knowledge in terms of text and figures. Automatic methods have been proposed to access to biomedical images and their associated metadata using text-based approaches [5, 7, 53] and content based methods [2, 24, 40, 42]. Considering the amount of images available in scientific literature, it is important that these methods can access the information in an efficient way. Difficulties related to scientific literature data include the heterogeneity of the images, the presence of compound images and the automation of ground truth labels extraction from the text [39]. The performance of the methods can be improved with pre-processing operation on the data (e.g. classifying the images according to their modality) or with post-processing operation on the results (e.g. using criteria for filtering the results). As stated by Müller *et al.* in [39] the PubMed Central (PMC) repository[9] includes more than 6 million articles in total with an average of 3.5 figures per article, including 1.5 compound figures of 4 sub-figures each. Their characteristics raise some open challenges [41, 65]. The sub-figures within a compound image usually represent different concepts, therefore it is necessary to separate them in order to apply content based image analysis methods [65]. Similarly, the caption also includes a textual description for each of the images. Hence, also in this case, the content related to its subparts needs to be identified and to be associated to the sub-images. The challenging aspect of the compound figure separation task lies in the lack of any standard regarding the formatting of multi-panel figures and their captions.

1.2.2 Compound figure separation, a definition

The problem of separating compound figures can be decomposed in two main phases, namely panel segmentation and caption splitting [65]. Panel segmentation consists in separating the compound image into the sub-figures that compose it and associating the right label to each sub-figure. Thus, panel segmentation can be itself divided into two subtasks: panel splitting (division of the figure into sub-figures) and label recognition (localizing and identifying the labels present in the image). Ultimately detected panels and labels have to be matched into pairs. Hence, panel segmentation only tackles visual information. On the other hand, caption splitting consists in identifying within

[9] http://www.ncbi.nlm.nih.gov/pmc/

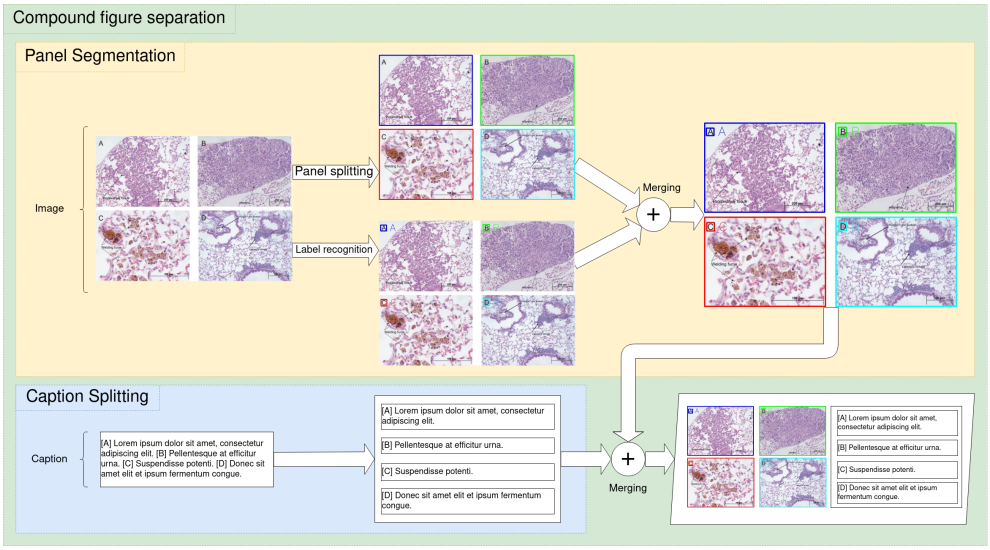


Figure 1.3: Overview of the compound figure separation task

the caption the textual information related to each sub-figure. Fig. 1.3 illustrates the subtasks hierarchy.

Despite this problem not being popular as a research task, previous works aimed at tackling multi-panel separation. More specifically, literature includes several works targeting one, or more rarely, a few subtasks of compound figure separation. However, papers describing the combination of the different phases together are much scarcer.

The compound figure separation task has the specificity of being multidisciplinary. It implies dealing with a combination of visual and textual information that share a common meaning. Extracting this precise meaning from each one of the sub-figures is the objective of this challenge. However, such information is not always easily splittable and the ground truth definition might be ambiguous.

For instance, the number of panels in a multi panel image can be unclear. In Fig. 1.4, one may annotate the three sub images on each of the two first rows as individual colors. However, the ground truth annotations distinguishes three panels, corresponding to the three rows of the image.

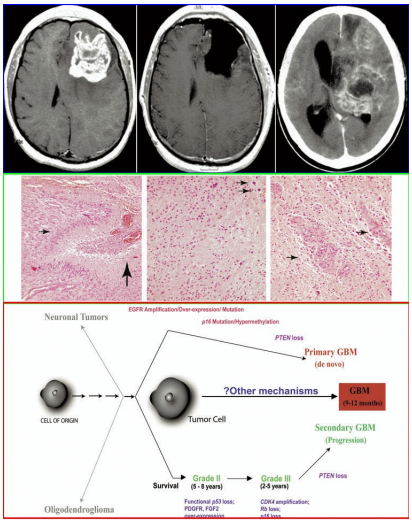


Figure 1.4: A compound figure with an unclear number of panels.
The three ground truth panels are delimited in blue, green and red.

On the other hand, compound figure captions are not easily splittable either. Some multi panel figures are described by a general caption describing the figure as a single unit. In other cases, the individual sub-captions are considered in a single sentence that could not be separated without loosing its meaning. An example of this type of situation is reported in Fig. 1.5.

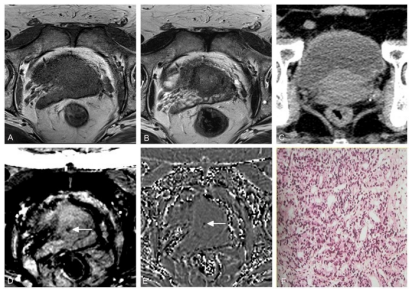


Figure 1.5: A compound figure with a caption that is hard to split:
`A 55-year-old man with prostate cancer in central zone of the prostate. No tumor hemorrhage is demonstrated on conventional T1WI (A), T2WI (B) and CT (C), but low signal within tumor on SWI (D) and filtered phase image (E) (arrows) indicates tumor hemorrhage. Histopathologic examination confirmed the diagnosis of prostate cancer (F).`

The final output of a compound figure separation algorithm is, given a compound figure image and its caption, the individual sub panels (images) linked to their respective sub caption.

1.2.3 Introduction to the proposed solution

This work adresses the problem of compound figure separation in its most general setting. Both visual and textual information are processed and more specifically combined by a unified pipeline. This thesis exposes a method and some results while being backed up by an open source code base. The latter was made publicly available and was written to ensure its usability. The further works foreseen in this area might benefit from this functioning baseline.

Panel segmentation The panel segmentation task (i.e. panel splitting and label recognition) was achieved using a modern deep learning architecture for object detection. The latter was adapted to this specific problem. The architecture design as well as the training process are developed in 3.1.

Caption splitting Dealing with the caption splitting problem involved designing specific algorithms that stand as original contributions to the field. Their implementation was carried out within the global pipeline and was quantitatively tested on a real data set.

Software engineering The implementation of the pipeline stands as an important part of the work. Designing a usable and robust software was one of the main objectives of this project and constitutes one of the final deliverables of the ExaMode project.

Finally, the main contribution of this work consists in uniting several state of the art solutions to achieve each subtask. Furthermore, innovative merging algorithms were designed to combine the results from the sub blocks.

1.3 Data sets

This project led to the manipulation of different medical data sets. At a granular level, the data processed by the pipeline consist in (compound) figure images from medical publications.

The **PubMed** database was made available in 1996. It offers more than 30 million references to biomedical and life science journal articles back to 1946. It was built by the National Library of Medicine[10] (NLM).

PubMed Central is a free archive for full-text articles released in 2000 [37]. It gives an open access to its complete articles, including their figures. The NLM collaborates with various publishers and journals to enrich their data base with updated content.

This latter data set of raw figure images was used to create material for compound figure separation tasks

[10] https://www.nlm.nih.gov/

- Panel splitting was proposed in the 2013 [17], 2015 [23] and 2016 [18] editions of the ImageCLEF medical challenge. The organizers provided annotations for panel bounding boxes for a few thousand images extracted from the PubMed central data set.
- Zou *et al.* addressed the panel segmentation problem in [65]. To evaluate the performance of their pipeline, the author built a data set with both panel and label annotations. The original images also come from PMC. This data set will be referred as *PanelSeg*.
- Concerning caption splitting, no data set has yet been made publicly available. Niccolò Marini and Stefano Marchesin built a collection of 250 annotated caption splitting examples. Each split caption comes with the associated picture.

1.4 An overview of this thesis

The following chapters describe in detail the different aspects of this work. In Chapter 2 are presented the previous efforts that have been made towards compound figure separation and its subtasks. The original methods, algorithms and the overall pipeline are exposed in Chapter 3. Chapter 4 discloses experiments conducted in various settings to test globally and individually the pipeline. In Chapter 5 offers a reflexion on the limitations and foreseen improvements. Finally, Chapter 6 draws conclusions on the study.

Chapter 2

Literature review

2.1 Object detection

The compound figure separation problem implies dealing with visual data (the multi-panel images). Panel segmentation involves the detection of both labels and panels within an image. As no clear standard exists for formatting multi panel figures, the splitting of those figures is a challenging task. As for several similar computer vision problems, the evolution of the image processing methods has let significant progress in terms of potential performance. In this first section, the most important computer vision techniques for object detection will be presented independently of the scope of this thesis.

Object detection is a central computer vision problem that has been explored for a very long time. The goal of this task is to detect objects within a 2D image and classify them into a predefined set of classes. The expected output of an object detection pipeline is a set of bounding boxes and, for each one, the probability of the detected object to belong to each class. First popular object detection pipelines were relying on traditional computer vision hand-crafted features as well as basic learning algorithms. Deep learning techniques relying on the famous convolutional neural networks drafted by Yann Lecun, Geoffrey Hinton and Yoshua Bengio in 1998 [27] stand now at the core of the computer vision domain.

Famous challenges have been designed for researchers and industrial actors to test their solutions on. The Pascal Visual Object Classes Challenge 2007 (VOC2007) [11] included an object detection challenge with 20 classes. Their data set was the main benchmark for any object detection system for a long time. More recently, the Miscrosoft COCO (Common Objects in Context) challenge [32] provided a more difficult task embodied by a larger data set. It includes over 200k labeled images containing objects from 80 categories.

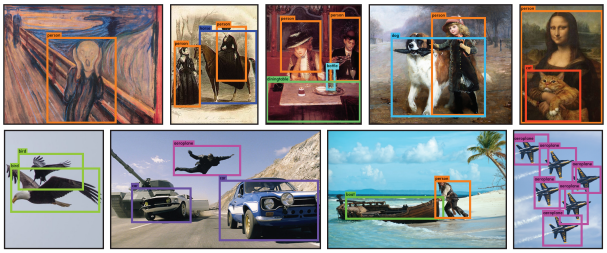


Figure 2.1: Example of predictions outputted by the YOLO [46] object detection system.

2.1.1 Traditional computer vision techniques

Despite the recent explosion of deep learning based algorithms for computer vision, traditional computer vision have long been developed to solve the object detection problem. O' Mahony *et al.* compared in a recent work [43] the deep learning and traditional approaches for computer vision. Several solutions considered as traditional rely on learning algorithms. However, those techniques differ significantly from modern deep learning convolutional neural networks.

Hand-crafted features Historic computer vision techniques involve, for the most part, hand-crafted features. Deterministic computations are done to extract relevant feature descriptors from images. They offer semantically rich and compact information that can then be processed to achieve a specific task. They are also called interest points because they often favour edges or corners which carry significant information.

The Harris detector leads to affine-invariant interest points and thus allows robust detection of key points across images taken from several view points. The underlying concepts of scale-invariant the Harris-Laplace Detector are the Harris measure and the Gaussian scale space representation. More precisely, the eigenvalues of the autocorrelation matrix A are interpreted to assess the importance of a point. The corner response function (Eq. 2.1) represents the interest of a point in terms of "cornerness".

$$R = \det(A) - k\,(Tr(A))^2 \tag{2.1}$$

After region points have been identified using this detector, they are filtered thanks to thresholding the corner response function. Finally, only local maxima of R are kept to yield the Harris key points. The Harris-Laplace detector is applied to multiple scales to obtain the characteristic scale. This process was proposed by Mikolajczyk and Schmid in [38].

The famous SIFT (Scale Invariant Feature Transform) detector introduced by David G. Lowe in [34, 35] allowed efficient object detection. The scale invariance property lets defining vectors able to characterize visual information independently from their size. SIFT computation is a multi-scale process that involves the following steps:

- **Scale-space extrema detection.** The idea is to find potential location for finding features. A scale space of images is created by constructing a set of progressively Gaussian blurred images. The difference of Gaussian kernel is applied to generate another pyramid.

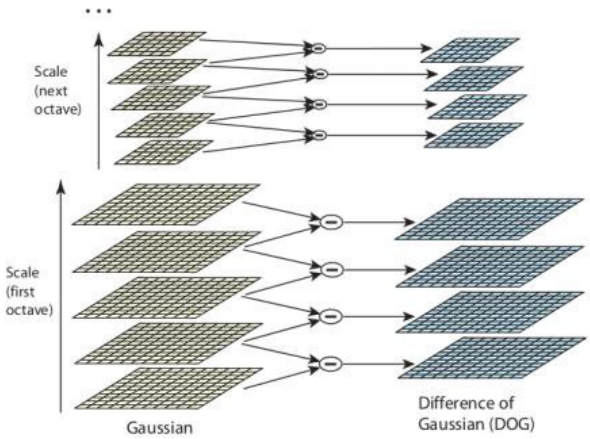


Figure 2.2: Difference of Gaussian from a blurred images pyramid, David G.Lowe [35]

Each pixel is then compared to its 8 neighbors as well as the corresponding 9 pixels from other scale images. The local extrema are kept as potential keypoints.

- **Keypoint localization**. The quadratic Taylor expansion of the Difference of Gaussian is used to interpolate the location of the extrema. Moreover, keypoints with low contrast are discarded. Finally, the keypoints corresponding to edges are suppressed in a similar way as in the Harris corner detection process.
- **Orientation assignment**. This step consists in assigning to each keypoint candidate an orientation in order to ensure rotation invariance. The gradient magnitude and orientation of each pixel in the keypoint neighborhood are computed to yield a histogram. Both the highest peak in the histogram and all peaks above 80% are used to compute the overall orientation.
- **Keypoint descriptor.** Keypoints have a location, scale and orientation. This step assigns to each keypoint a description vector built to be as independent as possible from viewpoint and illumination. A 16×16 window around the keypoint is divided into 16 sub-blocks of 4×4 pixels. An orientation histogram is computed. To avoid rotation dependence, the keypoint rotation is subtracted from each orientation to obtain a relative orientation. The vector is finally normalized and thresholded to eliminate dependence to illumination.
- **Keypoint matching.** The final step is used to match keypoints from two images. This process relies on finding the nearest neighbors of each keypoint.

Figure 2.3: Illustration of the extraction of SIFT descriptors in an image.

Similarly, the HOG (histogram of oriented gradients) features proposed by Navneet Dalal and Bill Triggs in [10] are also amongst the most used representations for various computer vision tasks. To achieve object detection, hand-crafted features are extracted before being fed to a simple learning model responsible for classification. See for instance Fig. 2.4. Support Vector Machine [9] or AdaBoost [15] are common examples of such classification tools.

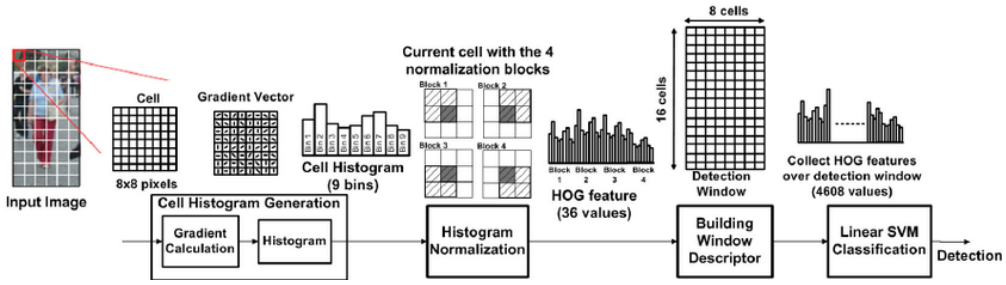


Figure 2.4: Object detection pipeline proposed by Suleiman *et al.* in [54].

Several other feature descriptors exist such as SURF [4] (2006), DAISY [57] (2010) and LIOP [60] (2011).

Viola-Jones object detector Paul Viola and Michael Jones proposed in 2001 [59] the first object detection framework. This work was focusing on the human face detection problem. The main benefits of this algorithm were its robustness and rapidity for the time (15 frames per second).

The algorithm has four stages [13]:

1. **Haar feature selection:** Haar features tend to benefit from the similarities that all faces have (for e.g. eyes are darker than the tip of the nose). They target the detection of edges, lines and diagonal features within the image. Given small rectangles, sum of pixels are computed and compared to assess the presence of a specific feature.

2. **Creating an Integral Image for fast feature computation:** Computing many Haar features can become computationally expensive. To avoid computing pixel differences for each possible 24x24 window, the authors proposed a concept of integral image. Each pixel of the original image is turned in a 2D cumulated sum of the pixels above it and at the left of it. Like so, obtaining the sum of pixel of a rectangle can be achieved by looking at the four corner values in the integral image. As the latter is computed in a single pass, the computation cost is greatly reduced.
3. **AdaBoost training for feature selection:** The AdaBoost algorithm [15] aggregates multiple weak learners to form a more capable strong learner. In this application, each Haar feature leads to a weak learner. The AdaBoost algorithm learns which predictors are performing the best and prioritize their vote in the final decision.
4. **Cascading Classifiers for fast rejection of windows without faces:** To avoid running the thousand classifiers on every region of the image, the classifiers are cascaded. The goal is to quickly discard non-faces and thus save computation time. The image subregions are sent through each classifier which gives a binary response for the presence of a face. If all the classifiers sequentially agree on a positive response, the window is said to contain a face.

The Viola-Jones pipeline gives a fair example of how a traditional object detection pipeline operates.

2.1.2 Deep learning

Computer vision features among the domains that have benefitted the most from the renaissance of deep learning. The AlexNet architecture [26] letting Krizhevsky *et al.* claim the ImageNet challenge win in 2012 marked the beginning of a new excitement for deep convolutional neural networks. Since then, more and more problems have been explored using deep neural networks. The latter offer the possibility to learn indirect visual characterization that tend to be optimal with respect to the objective task.

Three fundamental aspects of an object detection solution are:

- **The backbone:** Its role is to extract visual features from the image. Those can be compared to previously hand-crafted descriptors from older techniques. This component involves convolutional layers. As the backbone is a trainable model, the visual features it outputs are optimized for a given task. Object detection benefits from advances in other computer vision tasks such as supervised image classification. Convolutional feature extractors are often first tested on visual recognition challenges such as ImageNet [51] before being included in object detection pipelines.
- **The network architecture:** The model design is the core of an object detection solution. Several ways of exploiting visual features to finally obtain bounding boxes and class logits have been experienced. The use of fully connected layers have progressively been replaced by convolutional layers to give birth to fully

convolutional networks (FCN). This aspect of object detection systems includes the choice of the loss function. For instance, Lin *et al.* proposed an innovative loss in [31] which lead their RetinaNet system to beat state of the art models.

- **The training process:** Last but not least, as proposed model architectures are getting more and more complex, pipeline designers have come with innovative training processes. For instance, the Faster R-CNN architecture [49] required an alternating training protocole which differs from the training of single shot detectors such as YOLO [46]. The choice of training hyper parameters such as learning rates, number of epochs and batch size can play a significant role in final results. Scientific papers tend to always include those values and discuss them to ensure more reproducible research.

Following are some important and famous object detection systems that rely on deep learning architectures. Two main categories of object detectors have been experienced. One-stage detectors (YOLO [46], SSD [33]) use a single network for both region proposal and region classification. On the other hand, two shots detectors or region based networks split the region proposal tasks from the classification one.

2.1.2.1 Double shot detectors

R-CNN Girshick *et al.* proposed in 2013 their "Region-based convolutional networks" to address object detection. Regions of interest are first extracted by the selective search algorithm based on the heuristic that similar pixels usually belong to the same object. It favours regions that are similar in color, texture, shape and size. Those category independent regions are thus likely to contain meaningful information. They are fed into a large convolutional neural network (CNN) that turn those image patches into visual features. Finally, class specific support vector machines classify the proposed features. As region proposal from the selective search algorithm do not provide precise location information, the authors added an additional bound box regressor.

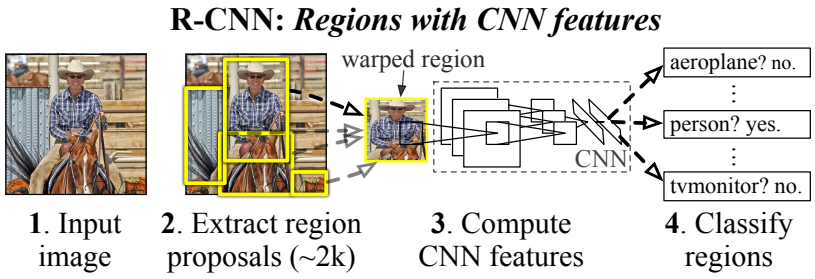


Figure 2.5: R-CNN pipeline [20]

Although this solution stood as the state of the art when it was proposed, it presents some drawbacks. On the hand, selective search not being a trainable algorithm denies the pipeline from being end to end trainable. At first, the CNN is trained on object proposals. The SVMs have then to be fitted to the extracted convolutional features. Finally, the

bounding box regressors are fine-tuned. This whole process is both challenging to implement and slow to operate as the CNN forward pass is done for each object proposal. On the other hand, the inferring process is computationally expensive both in space on time. The object proposals also need to be rescaled to fixed resolution and ratio.

Fast R-CNN To address the different flaws of the R-CNN architecture, Ross Girshick proposed an enhanced version named Fast-RCNN [19]. With this update, convolutional features are not redundantly computed anymore. Indeed, the whole image is given as input to the convolutional feature extractor once.

This architecture uses a variation of spatial pyramid pooling (SPP) to transform the visual features which have various sizes into fixed length vectors. The adaptation of SPP adopted in Fast R-CNN only has a single pyramid and is called the ROI (region of interest) pooling layer. The latter are also denoted as "bag of visual words" which refers to the similar procedure employed in textual information processing. This formatted representation allows the use of fully connected layers that splits in two output heads to perform both logits prediction and bounding boxes regression. For training, Fast R-CNN uses a multitask loss which is the sum of the classification loss and the L_1 loss (regression loss).

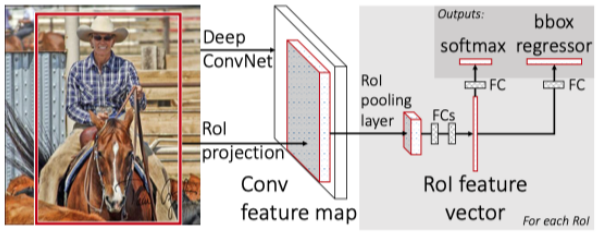


Figure 2.6: Fast R-CNN architecture [19]

Fast R-CNN leverages the main drawbacks of R-CNN by allowing and end to end trainable model. The unique forward pass of the convolutional neural network has allowed the inferring process to perform 10 to 20 times faster. However, during training, as the receptive field of a region of interest can reach the entire image, computations are expensive.

Faster R-CNN Even though Fast R-CNN tackled many of its ancestors limitations, it was still composed of the same untrainable component: the selective search algorithm for region proposal. Faster R-CNN [49] was then proposed to further improve the previous detector performance.

Fundamental changes were made including swapping the original region proposal procedure for a trainable one. Indeed, the motivation was to escape from the dependency of an external hypothesis generation method. The objects are detected in a single pass with a single convolutional backbone.

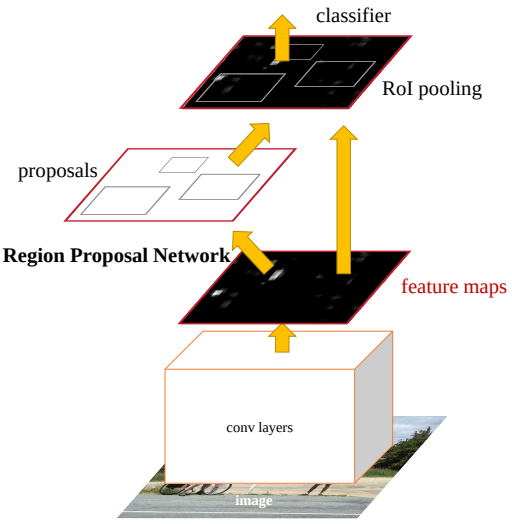


Figure 2.7: Faster R-CNN architecture [49]

The region proposal network (Fig. 2.8) is taking convolutional features as input. More precisely, a sliding 3x3 window is applied to the convolutional feature map. In practice, this is implemented thanks to a 3×3 convolutional kernel sliding on the feature map. This convolutional layer is followed by two sibling 1×1 used respectively for classification and regression. To deal with the variation of scales, the region proposal network relies on anchors. For each anchor location, k pre-defined prior boxes of different sizes and aspect ratios are considered as potential object detections. In the paper, the authors considered three different aspect ratios as well as three different scales which leads to $k = 9$ anchors types. The RPN is trained to discriminate the positive anchors which enclose an object from the negative ones which are background. It associates an objectness score to each anchor. The anchors are also regressed to fit as precisely as possible the contained object.

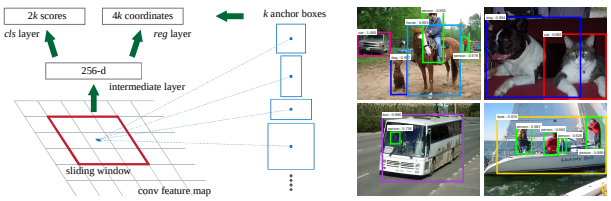


Figure 2.8: Faster R-CNN architecture [49]

The region proposals yielded by the RPN are then given as input to a ROI pooling layer (Fig. 2.7). From this point the architecture of Fast R-CNN is replicated. This final part is responsible for further regression and multi-class classification to output the final predictions.

The training of Faster R-CNN is achieved using a quadruple multitask loss:

$$\begin{aligned}\text{Loss} = &\underbrace{\text{RPN classification loss}}_{\text{good/bad anchors}} + \underbrace{\text{RPN regression loss}}_{\text{anchor} \rightarrow \text{proposal}} \\ &+ \underbrace{\text{Fast R-CNN classification loss}}_{\text{over classes}} + \underbrace{\text{Fast R-CNN regression loss}}_{\text{proposal} \rightarrow \text{box}}\end{aligned} \tag{2.2}$$

Faster R-CNN stands as a milestone in object detection. Its architecture was further refined in other works. The authors introduced a specific training process called alternating training. It consists in the RPN being trained first. The proposals are then used to train Fast R-CNN. Once the RPN has been fine-tuned while training Fast R-CNN, it is used to initialize RPN training. The process is iterated a few times.

2.1.2.2 Single shot detectors

You Only Look Once (YOLO) While two staged detectors where at the core of most of state of the art object detection pipelines, speed remained an issue. First, the training process was long and complex, but most importantly, the inferring speed did not allow real-time use cases. Joseph Redmon tackled this problem by designing a single shot detector that would accelerate significantly testing speed while limiting the impact on the accuracy.

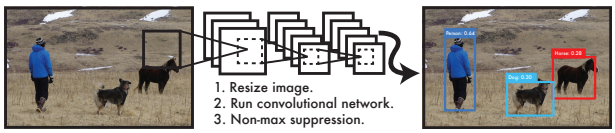


Figure 2.9: YOLO system, Joseph Redmon [46]

The YOLO architecture [46], being a one stage detector, does not feature a region proposal system. The object detection task is reduced to a regression problem which allows a simpler design. Both region proposal and classification are done simultaneously. A fixed grid of $S \times S$ cells covers the input image and each cell of the grid is the location of several boxes of different sizes and aspect ratios. The various sized regressors at each location output bounding box estimates (x, y, w, h) as well as one confidence score per box. This score represents the probability for an object to lie in this box. In the meantime, each grid cell provides the probability that the object belongs to each class. As both the grid size (total number of cells) and the number of boxes per location are fixed, the convolutional network that YOLO manipulates fixed sized tensors. This criterion is necessary for the use of the fully connected layers that process the convolutional feature vectors.

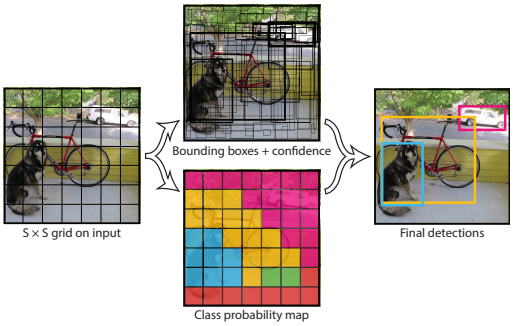


Figure 2.10: YOLO [46] fixed grid based detection

The convolutional network architecture is inspired by the GoogLeNet model [55]. The fixed grid design has limitations which consist in poor accuracies for smaller objects. The YOLO architecture leads to inferior but close average precision compared to Fast R-CNN. However, inferring speed reaches 45 frames per second which makes real time applications possible. The YOLO architecture was later updated [47, 48] to include more recent advances.

SSD: Single Shot MultiBox Detector After the YOLO architecture confirmed the relevance of single shot detectors, the SSD architecture tried to answer its main weaknesses. YOLO's fixed grid and boxes hypotheses limited its capacity to detect small objects. Moreover the possibility for several small objects to lie in a single grid cell was a clear flaw.

To tackle this challenge, Liu *et al.* presented the Single Shot MultiBox Detector (SSD) [33]. Its main specificity lies in the use of anchor boxes in a similar way as in the Faster R-CNN architecture. The management of multiple aspect ratio and scales was thus improved. Introducing anchor boxes made it possible to detect multiple small objects that lied in a single cell. Furthermore, it has let the network to differentiate overlapping objects more accurately. The image data first go through the truncated backbone convolutional network before traversing layers of decreasing size (Fig. 2.11).

This architecture creates feature maps of different sizes (Fig. 2.12). A map of size $m \times n$ gives rise to the same number of locations. At each locations, k anchor boxes are considered and corresponds to the output of c classes scores and 4 bounding box offsets each. Hence, for a single feature map a tensor of size $(c + 4)kmn$ is obtained.

Finally, the SSD paper presents different data augmentation methods such as cropping the image to obtain smaller patches or image expansion to simulate a zoom-out effect. The latter techniques simulate small objects from bigger ones while training. Hard negative mining is also used to preserve a ratio of approximately 3:1 between the number of negative and positive examples. Only the most challenging negative samples are kept to stabilize training. This improved single shot detector achieved state of the art results by outperforming the Faster R-CNN model on both PASCAL VOC and COCO data sets while being 3 times faster.

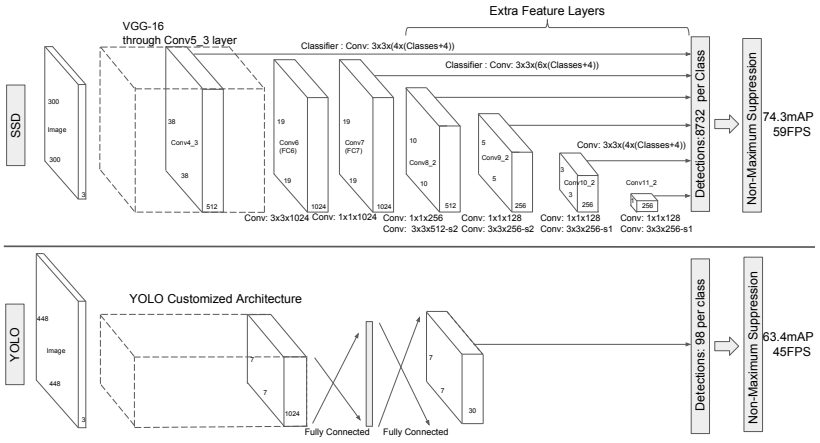
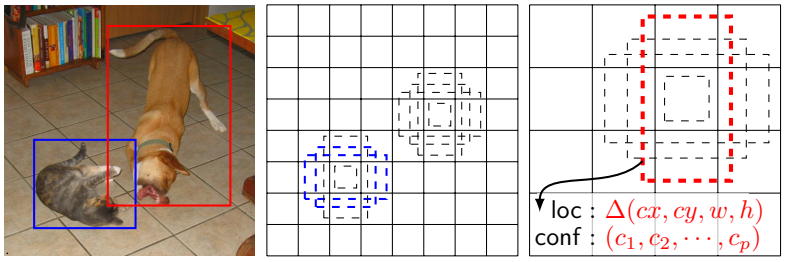


Figure 2.11: SSD network architecture (top) compared to the one used by YOLO [33]



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

Figure 2.12: Example of SSD feature maps. (Source: [33])

Focal Loss, RetinaNet One of the main weaknesses of one staged detection is the strong foreground background imbalance. Those models use a system of pre defined dense region proposal systems. Only a few of the million predicted boxes are accurately containing an object. YOLO proposed an intermediate classifier which attributed to each box candidate a confidence score to represent the likelihood of it actually localizing an object. SSD solved the problem differently by performing online hard negative mining to limit the imbalance impact on the loss. Neither of these ideas is perfect.

To address this drawback of one stage detectors, Lin *et al.* focused on the design of a new loss to train their RetinaNet detector [31] Their focal loss is defined as follow:

$$\mathrm{FL}(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t) \tag{2.3}$$

The focal loss is a generalized derivation of the tradition cross entropy loss: $\mathrm{CE}(p_t) = -\log(p_t)$. The added power γ (the focusing parameter) is used to balance the over sensitivity of the cross entropy loss to high confidence scores.

Besides the introduction of this new loss, the authors proposed the architecture of a novel single shot detector: RetinaNet (Fig. 2.13). It is composed of a Feature Pyramid Network (FPN) [30] that outputs feature maps of different scales. Classification and

bounding box regressions are each operated by a dedicated subnetwork. RetinaNet uses anchor boxes to ensure detections can be made at various scaled and aspect ratios.

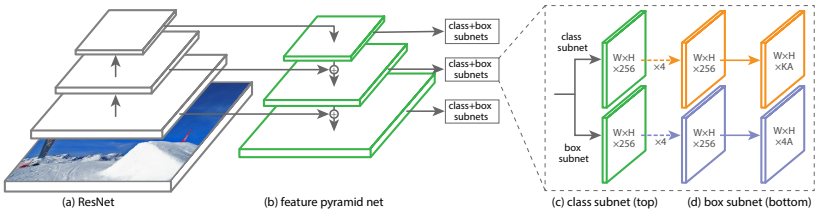


Figure 2.13: RetinaNet architecture, Lin *et al.* [31]

The RetinaNet architecture trained using the focal loss performed state of the art results on the COCO `test-dev` data set. It beats both top performing single shot detector DSSD (Deconvolutional Single Shot Detector) [16] and two stage Faster R-CNN [49]. In terms of detection speed, RetinaNet stands out with refresh rates around 30 frames per second.

Other architectures More recent advances have been offering higher performances in object detection.

- **YOLOv3** [48] (2018) is the last update of the previously presented YOLO architecture. It uses a more powerful backbone extractor combined with Feature Pyramid Network similar to the one employed in RetinaNet. This paper also demonstrated that focal loss offered a negligible advantage when using conditioned dense prediction. The industrial impact of this solution was compelling because of its simplicity, efficiency and speed.
- **Objects As Points** [63] (2019) discarded the anchor boxes in its CenterNet architecture. The network is in fact directly regressing the box height and width from its center. This process also lead to the disuse of the Non Maximum Suppression (NMS) algorithm which looses its relevance when not considering anchors.
- **EfficientDet** [56] (2020) focused on the FPN optimization. It includes a variant of a more recent FPN called BiFPN. The performance is increased thanks to the removal of certain useless connections as well as the addition of the weight feature fusion. Finally, the EfficientDet architecture has the advantage of being scalable and thus covers more use cases.

For the sake of brevity those solutions where not individually detailed. Jiao *et al.* [25] detail the most contemporary deep learning-based object detection techniques.

sectionPanel segmentation

2.1.3 Panel splitting

Panel splitting has been a recurring proposed task within the ImageCLEF[1] challenge [17, 18, 23]. As many other computer vision tasks, panel splitting was originally dealt with using traditional image processing algorithms before benefiting from more recent learning based approaches.

Traditional computer vision Older computer vision approaches usually exploit the gap between the different panels, using techniques adopted in detection tasks [28, 41]. Müller *et al.* [41] proposed an approach for panel splitting based on two phases: detection and analysis. In the detection phase, vertical and horizontal lines are identified, applying recursive operations to separate the panels. In the analysis phase, the lines classified as false positive are removed. Li *et al.* [28] developed a method based on connected Components Analysis (CCA) for removing small objects within an image. The algorithm maintains only the main components (the panels) inside the image. However, they noticed that the process is not working well with images that are blurred or not well-connected. Therefore, the author used an edge detector to improve blurred components detection and applied dilation on the edge image, in order to increase the connectivity between panels. Also in Cheng *et al.* [6], the panels are split using techniques for detecting vertical and horizontal lines. In this work, a pipeline of operations is applied for detecting the lines: a Sobel filter is applied to the image and then candidate bounding boxes are generated. A filter is then applied to remove false positive bounding boxes.

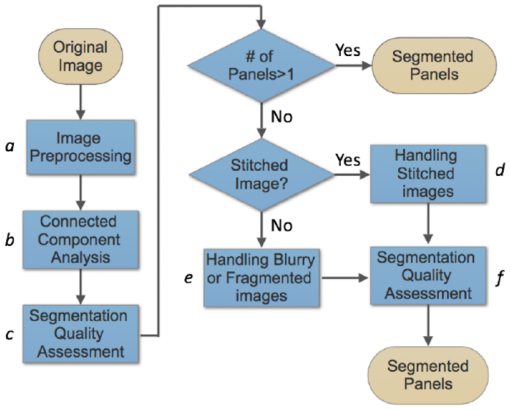


Figure 2.14: System for panel splitting proposed by Li *et al.* [28]

Deep learning based approaches The more recent approaches are based on machine learning algorithms and mainly exploit the results reached by Convolutional Neural Networks (CNN) in computer vision tasks [58, 64, 65].

In Tsutsui *et al.* [58], a CNN is trained for separating the panels. The problem is organized as an object detection task. The "You Only Look Once version 2 (YOLOv2)" system [47] is applied in order to detect the sub-figures and to define their bounding

[1] https://www.imageclef.org/

boxes (Fig. 2.15). This paper was the first, to the best of my knowledge, to apply a modern deep learning object detector to the panel splitting problem.

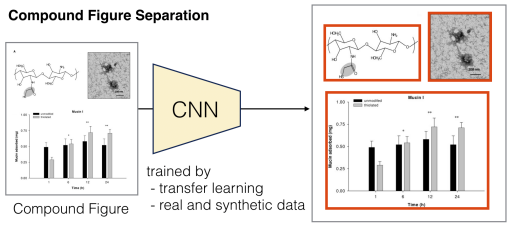


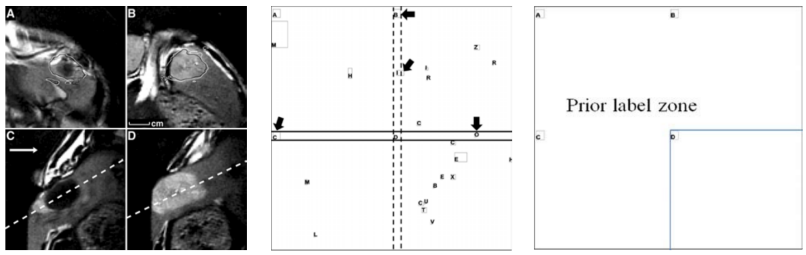
Figure 2.15: CNN-based panel splitting system introduced by Tsutsui *et al.* [58]

Besides using a recent architecture as the object detector, Tsutsui *et al.* boosted their results thanks to transfer learning and data synthesis. Indeed, the panel splitting data did not contained enough samples to properly initialize the YOLOv2 convolutional feature extractor. They instead benefited from pre initialized weights obtained while training the backbone for a classification purpose on the ImageNet data set [51]. This strategy was combined to sample synthesis. New compound images were automatically created by combining random figures in grid-like patterns with varying amount of white space for separating them. In so proceeding, the authors managed to boost their accuracy by more than 2% on the ImageCLEF 2016 data set.

In 2019, Zou *et al.* applied the RetinaNet [31] detector architecture to the whole panel segmentation problem. Their approach to deal with the whole task is explained below (2.1.5). They use the detector in their single class (panels) setting and trained it on both the ImageCLEF 2016 and their own PanelSeg data set (presented in the introduction 1.3). The ImageCLEF 2016 data set was used for comparing their results to panel splitting state of the art solutions. Thanks to the progress made on object detection by RetinaNet and its focal loss, the system performed better than the other existing techniques.

2.1.4 Label recognition

Label recognition was not the focus of many articles. You *et al.* [62] dedicated a paper to the sole task of label recognition. Their solution involved Optical Character Recognition (OCR) techniques. They handled false positive filtering by applying a custom belief propagation algorithm on a specifically defined Markov Random Field (MRF). Indeed, labels are more likely to be spatially ordered so the authors defined a neighborhood system (Fig. 2.16) for labels in order to fit in the MRF framework.



(a) Input (b) Neighborhood system (c) Prior label zone and detected panel labels

Figure 2.16: Illustration of panel label detection (You *et al.* [62])

Zou *et al.* [64] exposed a pipeline dealing with extracting labels from compound images. Their solution used an object detector similar to the ones described previously (2.1.1). They extracted a large set of HOG descriptors from the figure images that were then used to train a linear SVM classifier. The latter attempts at discriminating positive patches from background ones. To relax the precision-recall trade-off, they added a deep neural network for rejecting false alarms. Labels were then classified by feeding the filtered HOG features in a trainable RBF (Radial Basis Function) kernel SVM classifier [45]. The label recognition is thus approached as a classification problem of 50 classes (alphanumerical characters and digits). This work was the first one to identify label recognition as a multi-class object detection problem. This differs from the classical OCR approach and demonstrates the interpretability of the label recognition problem.

Nevertheless, the label recognition task is more meaningful when associated to panel splitting to become panel segmentation. Hence, Zou *et al.* [65] proposed a panel segmentation pipeline that extracted both panels and labels from figure images using convolutional neural networks.

2.1.5 Panel segmentation

Rarer works explored the panel segmentation task.

Zou et al. explored both subtasks of panel segmentation (panel splitting and label recognition) in [65], using a unified deep convolutional neural network. They modified the original RetinaNet architecture [31] to achieve the parallel detection of both panels and labels using a common convolutional backbone (Fig. 2.17). The single backbone is followed by two subnetworks that deals with respectively panel splitting and label recognition. The model works in this case in a single step, separating the figure in panels and detecting their labels simultaneously.

The panel and label features each go through a Feature Pyramid Network before being regressed and classified. They are then matched so that each panel is paired with a label using a beam search algorithm (Fig. 2.18). The overall panel segmentation problem (detecting both panels and labels) is expressed considered as pair of single class, respectively multi-class object detection tasks. The panel label matching is an additional step that has to be done separately.

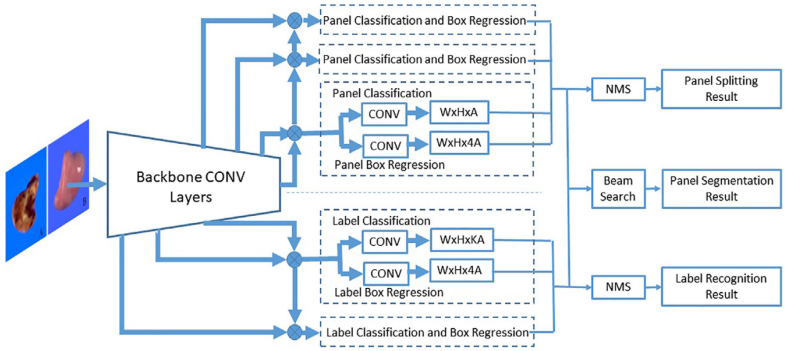


Figure 2.17: Unified architecture for panel segmentation (panel splitting + label recognition), Zou *et al.* [65]

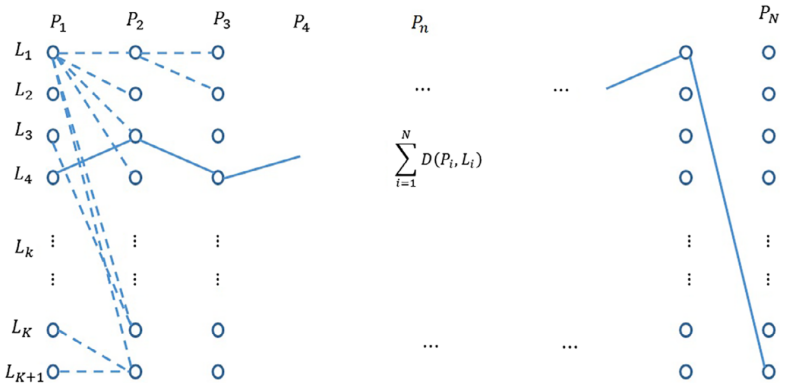


Figure 2.18: Illustration of the beam search algorithm, Zou *et al.* [65]

2.2 Caption splitting

Caption splitting can be considered as a subtask of the information extraction domain. Two main kinds of approaches are described in literature to perform caption splitting: approaches based on hand-coded rules and approaches based on machine learning methods.

Hand-coded rules techniques consist in applying a set of specifically crafted decision processes to the caption. Cohen *et al.* [8] apply a set of hand-coded rules in order to extract and classify image pointers from caption. They propose two methods for the evaluation. Each of the image labels is classified into three classes: bulleted list indicators, proper noun indicators and reference indicators. Two hand-coded approaches are tested. The first method achieves high precision (98.5%) but very low recall (45.6%), while the second achieves lower precision (74.5%) but higher recall (98.0%) and a higher F-score. Apostolova *et al.* [3] adopted similar rules for their work, but focusing on the extraction of bulleted list indicators. In both the previous works, false positive labels are eliminated using filter rules. In [1], a semi-automatic method, based on different hand-coded rules is proposed. In this work, the algorithm needs the image label type (e.g. `(A)`, `A)`, or `A:`)

as input. The information is used in order to extract the image labels and the sub-figure captions from the figure caption.

Freitag and Kushmerick [14] introduce a machine learning method to learn string patterns, in order to split captions. However, the algorithm needs several input parameters: the starting index, the ending index and the length of each text label within the caption. The approach takes inspiration from [8], but differs practically.

Due to the lack of numerous manually annotated samples, the use of modern deep learning models is limited.

2.3 Compound figure separation

To the best of my knowledge, Apostolova *et al.* [3] proposed the only work considering the full compound figure separation problem as defined in Section 1.2.2. Both image and text information were processed and original merging methods were proposed. Note that in the pipeline diagram (Fig. 2.19), the authors swapped the definitions of panel segmentation and panel splitting with respect to the ones presented in Section 1.2.2.

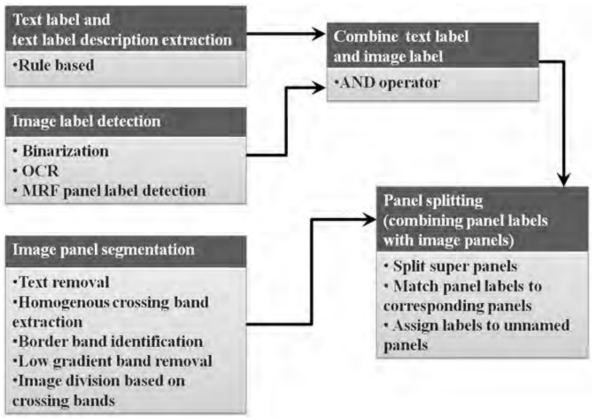


Figure 2.19: Process diagram showing contribution of each step to the multi panel figure segmentation algorithm, Apostolova *et al.* [3]

Their solution used for panel splitting (Fig. 2.20) relies on traditional computer vision techniques. The major steps are:

1. image overlay/markup removal,
2. homogeneous crossing band extraction,
3. border band identification,
4. image division based on crossing bands.

This pipeline boils down to a deterministic hand-crafted process and is highly dependant on the formatting of the figures. Examples which are not following a grid-like structure are more likely to be incorrectly split.

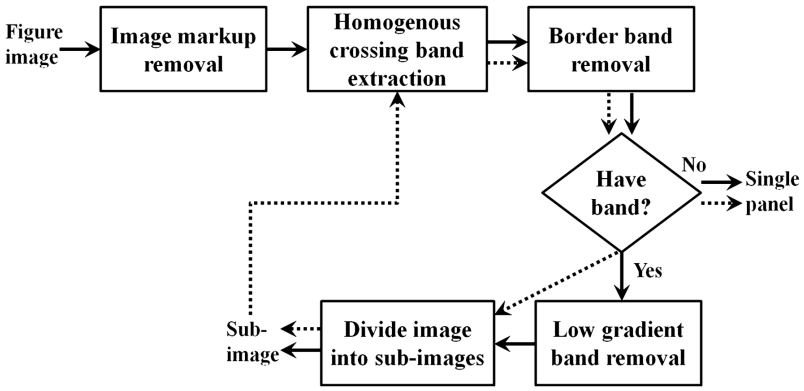


Figure 2.20: Panel splitting pipeline, Apostolova *et al.* [3]

The exposed label recognition solution is using an OCR technique on the preliminarily pre-processed image. The detection is improved in a similar way as in [62]: labels positions should be spatially coherent and have to respect a grid-like pattern.

To achieve panel segmentation, a 3 step process attempts to match detected panels and labels.

The authors did not publicly released their data set which prevents any form of quantitative comparison. The pipeline presented in this thesis differs from the one from Apostolova *et al.* [3] mostly by the technological choices within the different blocks. For instance, no form of deep learning techniques was originally used. Nevertheless, many interesting ideas are summarised by the authors of this paper.

Chapter 3

Methodology

In this chapter, the different elements of the proposed pipeline will be exposed.

The individual subtasks of compound figure separation are illustrated in Figure 1.3. On the image side, both panels and labels have to be extracted. These tasks are panel splitting and label recognition. Once the detected elements are matched into panel-label pairs, the results of panel segmentation is obtained. On the other hand, caption splitting consists in separating the caption in several sub-captions. Finally, the definitive results of compound figure separation is collected after both image sub-panels and textual sub-captions get matched into pairs.

This work presents a complete procedure to deal with all the mentioned tasks required to perform compound figure separation. Each step is detailed in the following subsections.

3.1 Panel segmentation

Panel segmentation targets both panel splitting and label recognition. The results from those two subtasks are paired thanks to a beam search algorithm.

First, the two subtasks of panel segmentation were solved individually by two independent neural networks. Second, the unified network introduced by Zou *et al.* was cleanly implemented and tested.

3.1.1 Panel splitting

Panel splitting consists in dividing the compound figure into the sub-figures that compose it. Panel splitting is achieved using the RetinaNet architecture [31] for object detection. This model is based on the *Focal Loss* for dealing with extreme foreground-background class imbalance. Labels are completely ignored when dealing with panel splitting.

For this specific task, the RetinaNet model was left untouched. The convolutional backbone is ResNet50 [22]. It would have been possible to change it for using its deeper version: ResNet152.

At test time, the raw output of the neural network goes through a Non Max Suppression (NMS) procedure. This approach is fairly common in modern object detection pipelines. It is used to filter out the overlapping predictions made by the model.

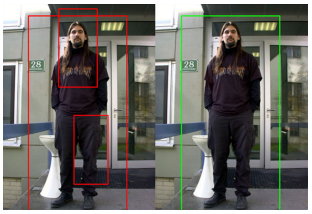


Figure 3.1: Before (left) and after the NMS algorithm was applied (Source: [50])

It is important to note, though, that NMS was configured specifically for this task. Indeed, contrary to conventional object detection in photos, panels will almost never overlap with each other. By choosing a significantly low IoU (Intersection over Union) threshold, predictions involving overlapping proposals are rejected. Quantitatively, 0.5 is commonly chosen as a standard value for the NMS IoU threshold. This was here decreased as far as 0.1 to avoid any overlapping panel detections. This solution was chosen after experimental tests, since it lead to a noticeable improvement of performance.

3.1.2 Label recognition

The task identified as label recognition is independent from panel splitting. The goal is to localize and identify the labels possibly present in a compound figure.
Currently, this phase targets single character labels. Zou et al. [65] proposes 50 different classes representing single alphanumerical character labels. Letters that share a similar shape between their upper case and lower case forms are representing a single class. Here are the 50 classes: a, A, b, B, c (c, C), d, D, e, E, f, F, g, G, h, H, i, I, j, J, k (k, K), l, L, m, M, n, N, o (O, o), p (P, p), q, Q, r, R, s (S, s), t, T, u (U, u), v (V, v), w (W, w), x (X, x), y (Y, y), z (Z, z), 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

For sake of convenience, the same base network architecture as for panel splitting was used for this task. However, differences should be noted. On the one hand, the features used are not the same as the ones used by the panel splitting network. The architecture details are presented in the following paragraph (3.1.3). On the other hand, the classification head of the model is outputting probabilities for detections to belong to one of the 50 label classes.

For both panel splitting and label recognition, the learning rate was set to 10^{-5}. Two GPUs were used for training with a batch size of two images (one per GPU). 90,000 steps were performed and were enough for the models to converge.

3.1.3 Unified architecture for panel segmentation

To simultaneously detect panels and labels from a single compound image, a single CNN feature extractor is shared by the two subnetworks that specialize into either panel

or label detection. This architecture was previously described by Zou et al. in [65] and it is illustrated in Fig. 3.2. More precisely, a single ResNet-50 [22] feature extractor is shared as a common backbone. Then, two different subsets of the backbone output are feeding two different Feature Pyramid Networks (FPN). The convolutional features used are C3, C4 and C5 for panel detection and C2, C3 and C4 for label recognition. The ResNet-50 architectures offers several features reflecting different receptive fields. C2, C3, C4 and C5 denote the residual blocks that constitute the ResNet 50 architecture. The higher the number, the deeper the layer and so, the larger the receptive field. Hence, to detect the labels that are smaller, we use features from lower layers. The difference in the FPN is justified by the fact that panel labels are smaller than the entire panels, thus requiring lower scale features. A classification head and a regression head follow each FPN. The panel classification head is trained to binary classify the presence of a panel in each proposed region. The regression head is trained to discriminate over the 50 classes presented by [65] (see Section 3.1.2).

At this point, the presented model outputs, for a single image, a set of panel boxes and a set of labeled label boxes. However, a key step needed to achieve panel segmentation is to match the split panels and the recognised labels. In order to do this, we applied the beam search algorithm proposed by Zou et al. [65], which is a greedily approach aiming at matching the detected panels and labels. This algorithm also helps eliminating panel or label false positives as it outputs panel-label pairs. The operation of this algorithm was presented in 2.1.5.

The training strategy for the unified architecture differs from the one used for the previous, smaller networks. The learning was equally set to 10^{-5}. However, the number of steps needed to be significantly increased for the model to converge. It was set to $1,000,000$ steps. As the batch size was set to 2 so that each of the two GPUs was handling a single image at a time. This was equivalent to slightly more than 100 epochs. The ResNet50 backbone was pre trained on ImageNet leading to a pre initialization of the weights in a transfer learning fashion.

3.2 Caption splitting

Captions convey important information that help understanding each sub-figure. The caption splitting component separates the caption, linked to the compound image, into sub-captions. Each one gets linked to a different panel (sub-figure) of the image. This caption splitting component is composed by a sequence of operations and it was tested on a manually annotated data set. The presented solution includes three main elements: labels extraction, label filtering and sub-captions extraction.

3.2.1 Label extraction

Label extraction is the equivalent of label recognition for the textual information. The goal is to infer which labels are present in the caption. Regular expressions are used to

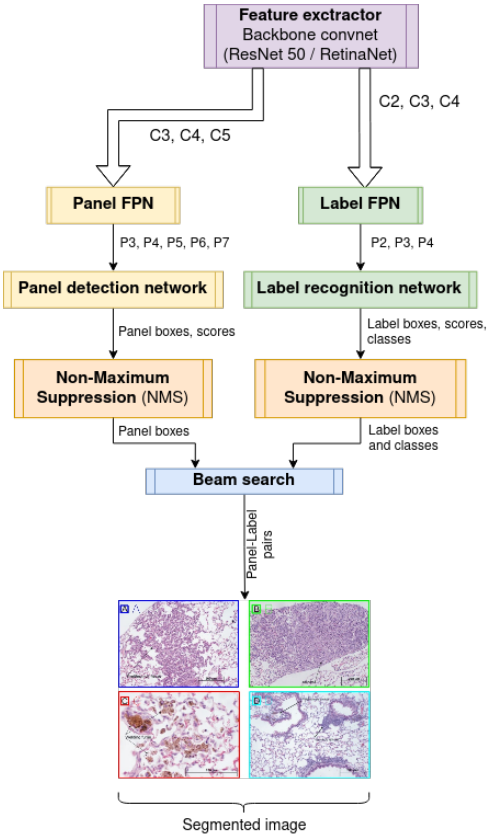


Figure 3.2: Unified neural network for panel label detection

achieve this operation. The different types of labels are: digits (*1*, *2*, *3*...), upper and lower case alphabetical characters (*a*, *B*, *j*,...) and upper and lower case roman numbers (*i*, *II*, *ix*,...). Matched regular expressions yield "positions". Three position classes are detected: 1. the labels that precede the panel description (e.g. "a) ..."); 2. labels that follow the panel description (e.g. "...(a)"); and 3. labels that are contained in a Part of Speech (POS) description (e.g. labels preceded by words like *in*, *from*, and *panel*). The labels classified as Part of Speech are not considered as actual labels since they are used for reference or as proper names within the sentences.

3.2.2 Label filtering

The detected labels are then filtered. Indeed, it is important to discriminate false positive (meaningful part of the caption) from true labels. In the context of a full compound figure separation (i.e. doing also the image panels and labels detection), both the labels detected from the image as well as the ones detected within the caption are merged before filtering. Proceeding as such takes advantage of both visual and textual aspects of the input data and allow a more robust detection (see 4.2). Once merged, the label merging algorithm identifies the most likely label structure. A label structure is defined by one type of label (upper case roman numbers, digits, etc.) and a number of labels.

Hence, even if detections lack some labels within a sequence or if some noisy labels have been detected, the final decision can still be correct. The process starts by building a histogram of the different label types. Each label votes for its type relatively to its index. We introduce a weight to account for how far the label is within this structure.

$$f(\text{label}, \text{type}) = \exp\left[-\text{index}(\text{label}, \text{type})\right] \tag{3.1}$$

index(label, type) represents the index of the label for this specific structure. For instance, index(D, *upper case letters*) $= 4$. This choice heavily penalizes labels that have high indexes and chooses the label type where the most low index labels have been detected. The label i for example would vote twice: A first time for the type *lower case alphabetical* with weight of $f(i, \textit{lower case letter}) = e^{-9} \simeq 1.2 \times 10^{-4}$ as the letter i is the 9th letter in the alphabet and a second time for the type *lower case roman number* with weight of $f(i, \textit{lower case roman number}) = e^{-1} \simeq 3.6 \times 10^{-1}$ as i accounts for the number 1. Outliers in detections are thus ignored by this process.

In a second step, the length of the sequence is inferred. The list of labels is truncated after the first index gap wider than two. As the output of this filtering process is a label structure, small gaps are thus ignored. For instance, the list `['A', 'B', 'D', 'W']` would yield to the detection of a label structure of upper case letters of length four (`['A', 'B', 'C', 'D']`).

3.2.3 Sub-caption extraction

Once a label structure has been outputted, the caption separation algorithm splits the caption into several sub-captions. A set of hand-coded rules is applied in order to fragment it in snippets. First, the preface gets detected. The preface sentence is the beginning part of the caption that will be common to each sub-caption. Even though it is not always present, this rule handles a fair part of the captions. Then, all the sentences following are processed one by one and assigned to one or more labels. Depending on the class, the text snippet contained between two labels is assigned to the preceding label (pre-description class) or to the following label (post-description class). If a text snippet is associated to label ranges or sequences, it is duplicated as many times as the number of labels in the range/sequence (e.g. the range A-D duplicates the text snippet for A, B, C, and D).

The example reported below shows the input and output of the algorithm applied to a caption. In this case, the labels belong to the post-description class.

```
    INPUT:
Immunohistochemical expression of c-MET in human prostate cancer. c-MET
is highly expressed in scattered prostate cancer cells (A), and
particularly at invasive fronts within peri-prostatic fat tissue (B);
```

```
arrowheads indicate positive cells.  Original magnification 100x.
```

```
OUTPUT:
```

- `A: Immunohistochemical expression of c-MET in human prostate cancer.c-MET is highly expressed in scattered prostate cancer cells. Arrowheads indicate positive cells.  Original magnification 100x.`
- `B: , and particularly at invasive fronts within peri-prostatic fat tissue.  arrowheads indicate positive cells.  Original magnification 100x.`

3.3 Compound figure separation

The full compound figure separation uses the different blocks described in the previous sections. The entire process is represented in Appendix A.

First, the figure image is fed to the custom unified RetinaNet model which tackles the simultaneous detection of both panels and labels. This step yields image panels and labels detections.

In parallel, the caption is processed to detect the labels from the textual information. Image labels and text labels detections are merged and filtered by the "Image-text label merging" algorithm. This step is an original contribution as it combines detections from to types of source data to better infer the latent structure of the compound figure. The output list of labels is then used to apply the sub-caption extraction algorithm which will split the caption text according to the detected labels.

Finally, both image panels and sub-captions are merged to output a set of sub-figures.

3.4 Software engineering, the `CompFigSep` library

Despite targeting a very specific problem, compound figure separation, the scope of this thesis is wide. Of course, as described in Chapter 2, several solutions were previously developed to partly solve this problem. However, hardly no work did tackle the task in its more general setting. Also, the majority of the existing solutions was not backed up by annotated data sets nor code. Hence, the motivation to write a complete program handling multi panel figure segmentation required an implementation effort (12.7k lines of code).

Open source software with improved modularity The MedGIFT team is part of the ImageCLEF challenge organizers and is convinced of the importance of the compound figure separation problem. Thus, for optimizing the positive impact of this work, the code was written in an efficient way from the beginning. The objective was not to limit

the implementation to scattered experiments or proof of concepts. The idea was rather to build a solid software library that would be later optimized and enhanced. Future interested researchers might use the developed pipeline to test their own algorithms and assess their performance. The `CompFigSep` code is presented on a public GitHub repository[1].

3.4.1 Used technologies and frameworks

The code is written in the Python language. This choice was made to favour portability and reliability. The majority of machine learning related implementations are developed in Python.

Deep learning frameworks The deep learning related algorithms are relying on the PyTorch library[2] [44]. The dynamic graph definition technology allows efficient testing and debugging cycles. Lately, the deep learning research community tends to shift from Google's TensorFlow API[3] [36] to PyTorch (notably exposed in [52]). The implementation backing up this thesis uses an intermediate layer to use PyTorch: FAIR's (Facebook AI Research) computer vision API, Detectron2[4] [61]. The latter gives a modular implementation of ubiquitous object detection models such as Fast R-CNN [19], Faster R-CNN [49], Mask R-CNN [21] and RetinaNet [31]. `CompFigSep`'s original models and training scripts where adapted from the Detectron2 tools.

Text processing The implemented solution for caption splitting relies mostly on regular expressions. The evaluation code, employs the powerful `textdistance`[5] API. The latter provides efficient implementations of various edit based, token based, sequence based and compression based text distances. The dedicated metric for caption splitting, which is detailed in Section 4.2, uses the implementation of the Levenshtein distance from this library.

3.4.2 Main components

Here is a very brief overview of the main original contributions regarding software implementation. Those technological solutions are generic and are an addition to the figure separation algorithms and offer them a basis to coexist.

The `Figure` object To propose an abstraction framework, an implementation of what a compound figure deeply is was developed. The `Figure` object encloses all the necessary methods and attributes for compound figures management (annotation loading, processing steps, preview functions, export tools, etc.). Implementation[6]

[1] https://github.com/GaetanLepage/compound-figure-separator
[2] https://pytorch.org/
[3] https://www.tensorflow.org/
[4] https://github.com/facebookresearch/detectron2
[5] https://github.com/life4/textdistance
[6] https://github.com/GaetanLepage/compound-figure-separator/blob/master/compfigsep/utils/figure/figure.py

Figure generators The `CompFigSep` library handles various formats for data which allows it to ingest different data sets (ImageCLEF, PanelSeg, etc.). An abstract concept of "figure generator" was introduced and allows the loading of the necessary data sets into Python generators of `Figure` objects. Implementation[7].

JSON export format A unique format was specifically created to serialize `Figure` objects to output files. This format gathers both ground truth and detected elements which allows performing evaluation independently from prediction. Implementation[8].

[7] https://github.com/GaetanLepage/compound-figure-separator/tree/master/compfigsep/data/figure_generators

[8] https://github.com/GaetanLepage/compound-figure-separator/blob/master/compfigsep/data/export.py

Chapter 4

Results

The pipeline to perform all the phases required to separate compound images was completed, tested and made publicly available on GitHub[1] and linked from the ExaMode web page[2]. The image related implementations rely on the Pytorch library [44] through Facebook Detectron 2 API [61]. The code for each subtask was tested independently, providing the results summarized below, in order to allow the comparison of performance with previous works.

4.1 Panel segmentation

4.1.1 Panel splitting

Data sets The goal of panel splitting is to localize the panels within a compound figure. This task can be described as single class object detection problem. It was first proposed within the ImageCLEF 2013 challenge [17] and reproposed within the ImageCLEF 2015 [23] and 2016 [18] editions. The model for panel splitting was evaluated on the 2016 data set to be able to compare the results to several recent works. An example of an output from the panel splitting detection network is reported in Fig. 4.1.

The ImageCLEF data was divided in the following way to train the neural network: 6783 samples (81% of the data set) were used for training and 1,614 for testing. The ImageCLEF 2016 data set is limited by including annotations for panels only. Hence, it could not be used to evaluate performance of the system for the other tasks (caption splitting, label recognition and thus panel segmentation). To overcome this limitation, Zou et al. [65] gathered a new data set that will now be referenced as the *PanelSeg data set*. The authors have extracted 10,642 figures from the original PubMed Central data set[3] and annotated both the panels and the labels. Evaluation of label recognition and panel segmentation tasks have then been conducted on this data set. The PanelSeg data were divided in the following way to train the neural network: 9,642 samples were used for training 90%) and the remaining 1000 images were used for testing.

[1] https://github.com/GaetanLepage/compound-figure-separator
[2] https://www.examode.eu/software/
[3] http://www.ncbi.nlm.nih.gov/pmc/

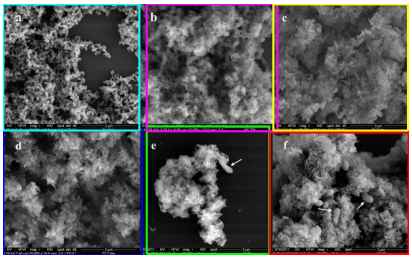


Figure 4.1: Example of panel splitting output

ImageCLEF metric The panel splitting task, as presented in the ImageCLEF 2013 challenge [17], is evaluated by a specific metric. This score does not correspond to the traditional object detection metrics (precision, recall and mean averge precision). The score for a single figure is defined as follows:

$$S_F = \frac{C^F_{\text{correct}}}{\max(K^F_{GT}, K^F_C)} \tag{4.1}$$

where:

- C^F_{correct} is the number of correct matches. For each ground truth panel bounding box, the best matching detection is found. Then, if the size of the ground truth box is at least 66% of the candidate's size, the match is said to be correct. Since only one candidate detections can be assigned to each of the ground truth panels, we have that $C^F_{\text{correct}} \leq K^F_C$.
- K^F_C is the number if candidate panels.
- K^F_{GT} is the number of ground truth subfigures.

The specific score normalization factor penalizes a too large number of detections and ensures the value cannot be greater than 1. The maximum score is reached by providing exactly K^F_{GT} detections that are all correct. We then have:

$$C^F_{\text{correct}} = K^F_C = K^F_{GT}$$

The scores of all figures are then averaged to obtain the ImageCLEF accuracy:

$$\text{Acc} = \frac{1}{N_{\text{figures}}} \sum_{i=1}^{N_{\text{figures}}} S_{F_i} \quad \in [0,1] \tag{4.2}$$

Results The results for the panel splitting task (on the ImageCLEF 2016 data set and on the PanelSeg data set) are reported respectively in Table 4.1 and in Table 4.2. The used evaluation metrics are precision, recall, and mean Average Precision (mAP). Since in the panel splitting task we only detect a single class (panels), the mAP metric corresponds to the average precision. The performance of the algorithm on the ImageCLEF data set

was also evaluated with the "ImageCLEF accuracy". The performance of the model proposed by Zou et al. [65], Tsutsui et al. [58] and Pengyuan et al. [29] are also provided.

Table 4.1: Panel splitting results (ImageCLEF 2016 data set)

Model	ImageCLEF accuracy	Precision	Recall	mAP
Tsutsui et al.	84.6	87.5	75.1	77.3
Pengyuan et al.	84.43			
Zou et al. (ResNet 152)	85.1	89.8	**78.9**	**78.4**
Zou et al. (ResNet 50)	83.8	**90.0**	77.7	78.6
Ours (ResNet 50)	**85.2**	88.2	77.4	75.8

Table 4.2: Panel splitting results (PanelSeg data set)

Model	Precision	Recall	mAP
Zou et al.	**82.9**	91.1	**88.4**
Ours	68.8	**92.0**	89.3

4.1.2 Label recognition

Label recognition was evaluated on the *PanelSeg* data set as it is the only one containing ground truth annotations for image labels. The results for the label recognition task are reported in Table 4.3. Also in this case, the used evaluation metrics are precision, recall, and mean Average Precision (mAP). The performance of the model proposed by Zou et al. are also provided. Even though our implementation mainly follows a very similar architecture, the results of the model developed in the context of this work are significantly better in terms of precision, recall and mAP.

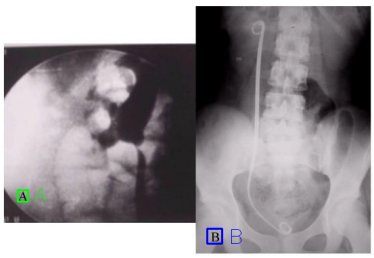


Figure 4.2: Example of label recognition output

Table 4.3: Label recognition results

Model	Precision	Recall	mAP
Zou et al.	12.4	85.9	55.0
Ours	**52.97**	**88.26**	**53.3**

4.2 Caption splitting

Metric A metric to evaluate this specific problem was not presented before in literature. Therefore, we decided to assess the performance by measuring the average normalized Levenshtein similarity. The Levenshtein distance is an edit distance. It gives an idea of how different two characters strings are based on the number of edits to go from one to the other. The following definition allows a recursive computation of the Levenshtein distance between strings a and b.

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j) + 1 \\ \mathrm{lev}_{a,b}(i,j-1) + 1 \\ \mathrm{lev}_{a,b}(i-1,j-1) + \mathbb{1}_{a_i \neq b_j} \end{cases} & \text{otherwise.} \end{cases} \tag{4.3}$$

By definition, this distance is bounded as the number of edits to turn string a into string b is at most the length of the longer string.

$$0 \leq \mathrm{lev}(a,b) \leq \max\left(|a|,|b|\right) \tag{4.4}$$

From this definition, one can define the normalized similarity:

$$S(a,b) = 1 - \frac{\mathrm{lev}(a,b)}{\max\left(|a|,|b|\right)} \tag{4.5}$$

where $\mathrm{lev}(a,b)$ is the Levenshtein distance between strings a and b.
$|.|$ denotes the length of a string.

By construction, this metric is bounded between 0 when difference is maximum and 1 when both strings are equal. The score for caption splitting of a single figure is given by:

$$\mathrm{score}(F) = \sum_{l \in L_F} \begin{cases} S(sc_{\mathrm{gt},l}, sc_{\mathrm{det},l}) & \text{if } \exists\, sc_{\mathrm{det},l'} \in SC_F \text{ s.t. } l' = l \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

where F is a compound figure, l is a ground truth label in L_F, the sets of ground truth labels for figure F. $sc_{\mathrm{gt},l}$ and $sc_{\mathrm{det},l}$ are respectively the ground truth and the detected sub-caption associated with label l. SC_F is the set of detected sub-captions.

Introducing this metric for caption splitting leverages the qualitative human evaluation proposed in previous works such as [3].

Data set and results Caption splitting is evaluated on a partition of the PubMed data set. It includes captions and the corresponding ground truth (a set of sub-captions that were manually annotated). The partition includes 196 (originally 250) samples and it comes without ground truth annotations, thus required to manually create it. Each sample within the partition includes a caption and the corresponding compound image. In order to evaluate the performance of the algorithm, the captions were manually annotated by a former member of the MedGIFT team. For each of the captions, a list of labels and the corresponding text snippets were identified. After the process of manual annotation of the text snippets, only 193 samples were selected to compose the partition. The partition originally included 250 samples, but 57 of the 250 captions were discarded because within it was not possible to identify the labels or the corresponding text snippets. The caption splitting task is challenging as the ground truth cannot always be easily defined. Indeed, several captions are not semantically splittable as they would loose their meaning. Compound figures sub panels can be strongly linked to each other and the caption might describe the compound figure as a whole. The proposed algorithm achieves an averaged score of 0.72 on the annotated subset of the data set.

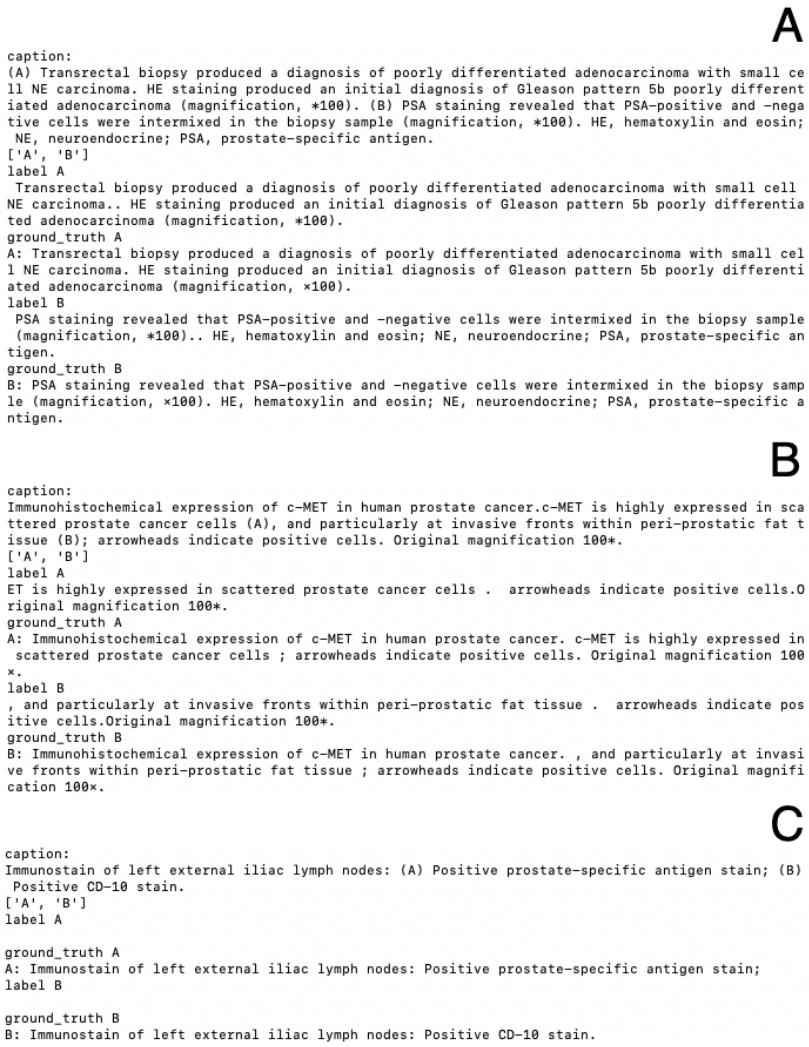


Figure 4.3: Example of caption splitting output. Panel A) includes includes an example of caption well split (labels equal to their corresponding ground truth). Panel B) includes an example of caption well split (labels similar to their corresponding ground truth). Panel C) includes an example of caption considered as not well split.

Chapter 5

Discussion

In this chapter, the extent of the obtained results will be analyzed. Also the limitations and shortcomings related to the proposed solution will be addressed.

Building a generic and complete pipeline for the compound figure separation problem has been a challenging task. First, the problem involves tackling several undertakings. Second, the software implementation efforts ended up being significant in terms of time resources.

5.1 Panel segmentation

Some unexpected outcomes happen when testing the different panel segmentation experiments.

Specific label recognition architecture As presented in Sec. 3.1.2, the RetinaNet architecture was adapted for the label recognition task. Zou *et al.* suggested to use different convolutional features to feed the classification and regression head of the label recognition subnet.

Unified architecture Despite very similar implementation and training strategy, this adaptation of the model from Zou *et al.* [65] did not perform well. Both panel splitting and label recognition results were considerably inferior to the performance hit with the individual networks. The training has let the model converge (see Fig. 5.1).

Backbone and detector model architecture As the panel segmentation task relies on object detection models, working with more recent architectures would certainly help achieving better performance. Important advances have been made in the computer vision field as the ones presented in Sec 2.1.2 would be worth trying out. The idea of the unified architecture can be adapted to any modern object detection system.

Data augmentation Tsutsui *et al.* [58] proposed a data synthesis technique to augment the quantity of training images. In the context of the panel segmentation task, figures might be indeed easily created from plain medical images. Artificially aranging the latter in a grid-like fashion would lead to likely compound figures.

To conclude on the image related tasks, further experiments and testing are necessary to make significant declarations regarding the results. The framework would benefit from more modern backbone architecture.

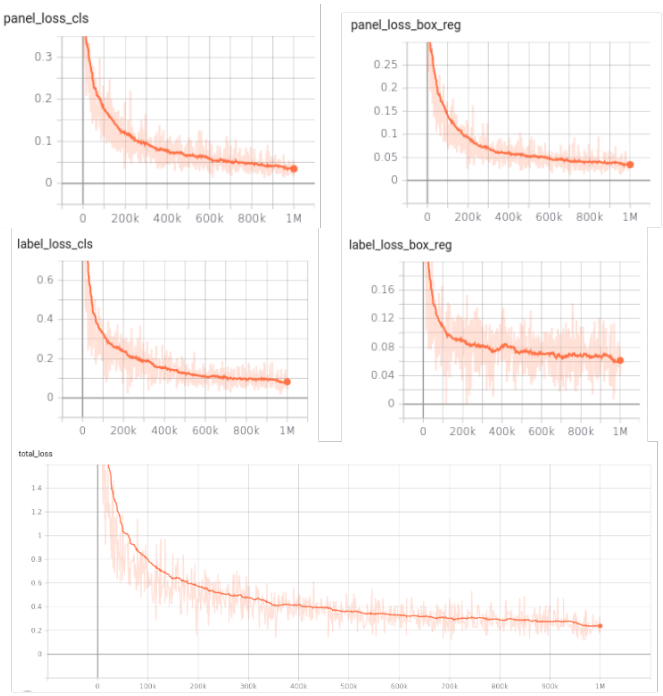


Figure 5.1: Training losses of the unified panel segmentation model

5.2 Caption splitting

The relevance of the caption splitting results is complex to assess. Indeed, as no public data set has yet been made available, no quantified results are waiting to be challenged.

Original contributions This thesis proposes original contributions to this task both in terms of algorithms and evaluation. The application of the normalized Levenshtein similarity as an evaluation metric will hopefully encourage future works to challenge the obtained results.

Augmentation of the PanelSeg data set for caption splitting An additional data collection task would be interesting to consider. Indeed, Zou *et al.* [65] have built a data set from raw PubMed Central images by annotating them with both panel and label bounding boxes. All the captions for those images were gathered from the PMC website during this project using a custom made script. The annotation process of the ground truth was not terminated and would make possible a complete evaluation of the full compound figure separation problem.

Chapter 6

Conclusion

Compound figure separation is a challenging problem that could have significant impact on medical information processing. Figures from biomedical scientific litterature offer a diverse range of settings, layouts and semantic information. Multi-panel figures are not directly processable by generic algorithms. Loosing a precious quantity of valuable data can be avoided thanks to separating algorithms.

This thesis defines and addresses the compound figure separation problem in its more general setting. The limited scientific litterature does not offer ready to use efficient solutions to this challenge. Hence, one of the objectives of the ExaMode project was to propose a generic tool for this task.

An extensive review of the existing partial solutions have been conducted. It led to the design of an overall solution aiming at combining the most advanced tools available for each subtask. The implementation of a complete suite of evaluation tools lets future works assessing the performance of their solution. A challenge similar to the ImageCLEF medical task could encourage the research community to embrace this problem and imagine new algorithms to tackle it. A universal benchmark backed up by an extensive benchmark is part of the motivations of this work.

Modern deep learning architectures have been specifically implemented to enhance the image processing module. The implementation of several solutions was integrated to the library and was able to match some state of the art results in panel segmentation.

The caption splitting task was by far the most unexplored task. It required the definition of a specific metric as well as innovative algorithms to address the challenge. Moreover, the proposed solution was relevantly integrated to the global pipeline in order to make the most out of the dual nature of the information. Indeed, the compound figure separation problem can be seen, in a more abstract way, as a more general problem. The goal is to infer the latent structure of a compound figure (through the detection of the labels from the text and the caption) to better isolate and match the image panels and sub captions.

This work stands as an innovative draft of what could be a fully working solution. Many optimizations and tests would have to be carried out to further improve the system.

Nevertheless, the pipeline design is highly compatible with many different algorithms. It has been thought from the beginning as an open library that encourages the change of any of its modules. The code has thus been made open source and is available on GitHub as a toolkit for reproducible research on compound figure separation.

Appendix A

CompFigSep pipeline diagram

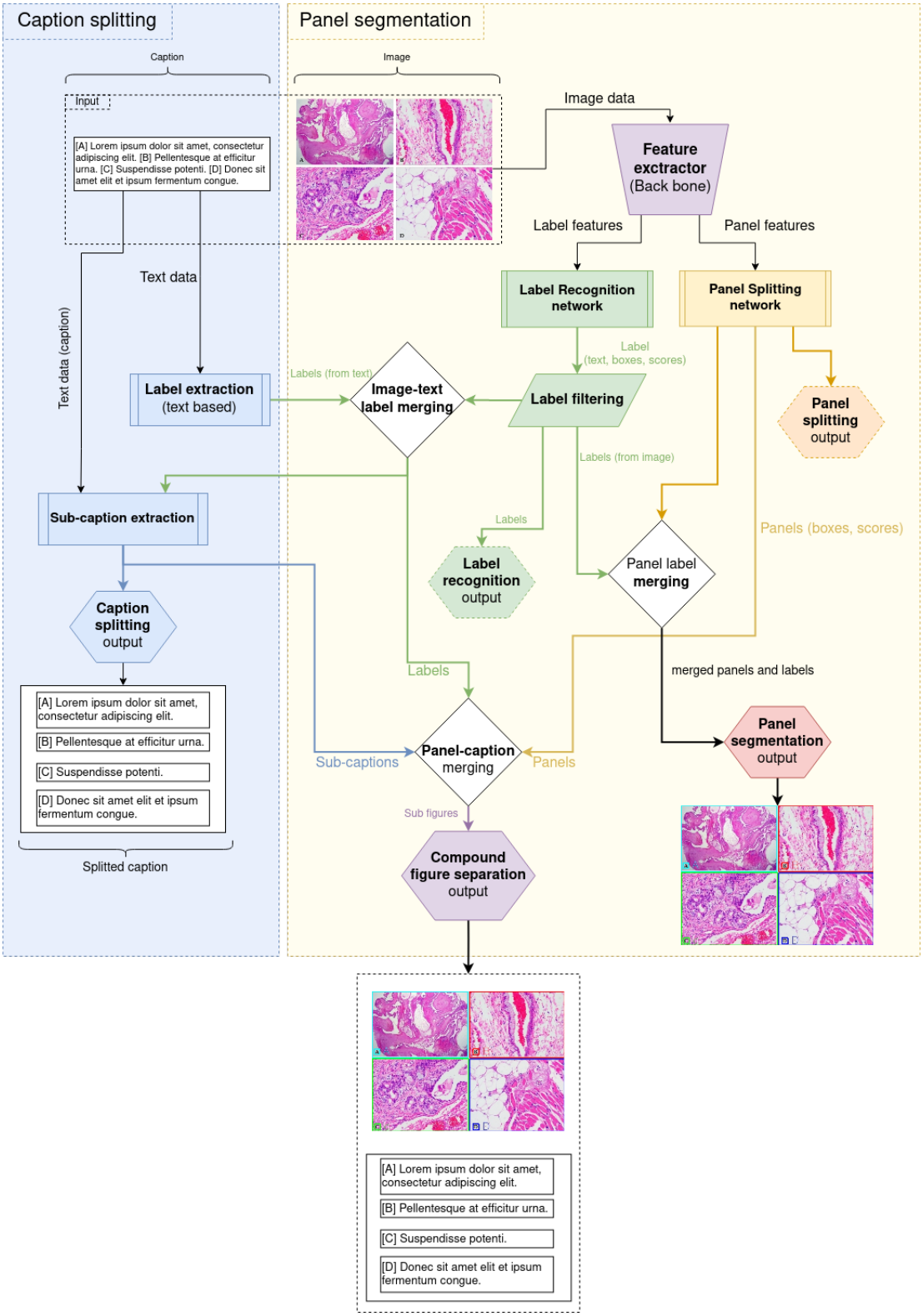


Figure A.1: The full CompFigSep pipeline.

Bibliography

[1] M. Ali, L. Dong, Y. Liang, Z. Xu, L. He, and N. Feng. "A novel algorithm for extracting text labels and subfigure captions from multi-panel figure caption". In: *2014 11th International Computer Conference on Wavelet Actiev Media Technology and Information Processing (ICCWAMTIP)*. IEEE. 2014, pp. 226–229.

[2] S. K. Antani, L. R. Long, and G. R. Thoma. "Content-based image retrieval for large biomedical image archives." In: *Medinfo*. 2004, pp. 829–833.

[3] E. Apostolova, D. You, Z. Xue, S. Antani, D. Demner–Fushman, and G. R. Thoma. "Image Retrieval from Scientific Publications: text and Image Content Processing to Separate Multi–Panel Figures". In: *Journal of the American Society for Information Science* 64 (5 2013), pp. 893–908.

[4] H. Bay, T. Tuytelaars, and L. Van Gool. "SURF: Speeded Up Robust Features". en. In: *Computer Vision – ECCV 2006*. Ed. by A. Leonardis, H. Bischof, and A. Pinz. Vol. 3951. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417. ISBN: 978-3-540-33832-1 978-3-540-33833-8. DOI: 10.1007/11744023_32. URL: http://link.springer.com/10.1007/11744023_32 (visited on 08/19/2020).

[5] K. W. Boyack, D. Newman, R. J. Duhon, R. Klavans, M. Patek, J. R. Biberstine, B. Schijvenaars, A. Skupin, N. Ma, and K. Börner. "Clustering more than two million biomedical publications: Comparing the accuracies of nine text-based similarity approaches". In: *PloS one* 6.3 (2011).

[6] B. Cheng, S. Antani, R. J. Stanley, and G. R. Thoma. "Automatic segmentation of subfigure image panels for multimodal biomedical document retrieval". In: *Document Recognition and Retrieval XVIII*. Vol. 7874. International Society for Optics and Photonics. 2011, 78740Z.

[7] A. M. Cohen and W. R. Hersh. "A survey of current work in biomedical text mining". In: *Briefings in bioinformatics* 6.1 (2005), pp. 57–71.

[8] W. W. Cohen, R. Wang, and R. F. Murphy. "Understanding captions in biomedical publications". In: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2003, pp. 499–504.

[9] C. Cortes and V. Vapnik. "Support-vector networks". en. In: *Machine Learning* 20.3 (Sept. 1995), pp. 273–297. ISSN: 0885-6125, 1573-0565. DOI: 10.1007/BF00994018. URL: http://link.springer.com/10.1007/BF00994018 (visited on 08/18/2020).

[10] N. Dalal and B. Triggs. "Histograms of Oriented Gradients for Human Detection". en. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. San Diego, CA, USA: IEEE, 2005, pp. 886–893. ISBN:

978-0-7695-2372-9. DOI: 10.1109/CVPR.2005.177. URL: http://ieeexplore.ieee.org/document/1467360/ (visited on 08/18/2020).

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[12] *ExaMode*. 2019. URL: https://www.examode.eu/ (visited on 08/18/2020).

[13] *Face detection as done in 2001: Viola Jones Algorithm*. en. Aug. 2019. URL: https://iq.opengenus.org/face-detection-using-viola-jones-algorithm/ (visited on 08/19/2020).

[14] D. Freitag and N. Kushmerick. "Boosted wrapper induction". In: *AAAI/IAAI*. 2000, pp. 577–583.

[15] Y. Freund and R. E. Schapire. "A Short Introduction to Boosting". en. In: (), p. 14.

[16] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. "DSSD : Deconvolutional Single Shot Detector". In: *arXiv:1701.06659 [cs]* (Jan. 2017). arXiv: 1701.06659. URL: http://arxiv.org/abs/1701.06659 (visited on 08/19/2020).

[17] A. Garcia Seco De Herrera, J. Kalpathy-Cramer, D. Demner-Fushman, S. Antani, and H. Müller. "Overview of the ImageCLEF 2013 medical tasks". en. In: ed. by P. Forner, R. Navigli, D. Tufis, and N. Nicola Ferro. Meeting Name: CLEF 2013 Conference. Valencia, Spain: CEUR Workshop Proceedings, Sept. 2014. URL: http://ceur-ws.org/Vol-1179/CLEF2013wn-ImageCLEF-SecoDeHerreraEt2013b.pdf (visited on 04/01/2020).

[18] A. Garcia Seco De Herrera, R. Schaer, S. Bromuri, and H. Müller. "Overview of the ImageCLEF 2016 Medical Task". en. In: ed. by K. Balog, L. Cappellato, N. Ferro, and C. Macdonald. Évora, Portugal: CEUR Workshop Proceedings, July 2016. URL: http://ceur-ws.org/Vol-1609/16090219.pdf (visited on 08/06/2020).

[19] R. Girshick. "Fast R-CNN". In: *arXiv:1504.08083 [cs]* (Sept. 2015). arXiv: 1504.08083. URL: http://arxiv.org/abs/1504.08083 (visited on 04/17/2020).

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik. "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation". en. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.1 (Jan. 2016), pp. 142–158. ISSN: 0162-8828, 2160-9292. DOI: 10.1109/TPAMI.2015.2437384. URL: http://ieeexplore.ieee.org/document/7112511/ (visited on 08/19/2020).

[21] K. He, G. Gkioxari, P. Dollár, and R. Girshick. "Mask R-CNN". In: *arXiv:1703.06870 [cs]* (Jan. 2018). arXiv: 1703.06870. URL: http://arxiv.org/abs/1703.06870 (visited on 08/21/2020).

[22] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[23] A. G. S. de Herrera, H. Müller, and S. Bromuri. "Overview of the ImageCLEF 2015 Medical Classification Task." In: *CLEF (Working Notes)*. 2015.

[24] W. Hsu, L. R. Long, S. Antani, et al. "SPIRS: A framework for content-based image retrieval from large biomedical databases." In: *MedInfo* 12 (2007), pp. 188–192.

[25] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu. "A Survey of Deep Learning-Based Object Detection". In: *IEEE Access* 7 (2019). Conference Name: IEEE Access, pp. 128837–128868. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2939201.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25* (2012). Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf (visited on 08/18/2020).

[27] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (Nov. 1998). Conference Name: Proceedings of the IEEE, pp. 2278–2324. ISSN: 1558-2256. DOI: 10.1109/5.726791.

[28] P. Li, X. Jiang, C. Kambhamettu, and H. Shatkay. "Compound image segmentation of published biomedical figures". In: *Bioinformatics* 34.7 (Apr. 2018), pp. 1192–1199. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx611. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6030860/ (visited on 04/01/2020).

[29] P. Li, S. Sorensen, A. Kolagunda, X. Jiang, C. Kambhamettu, and H. Shatkay. "UDEL CIS at ImageCLEF Medical Task 2016". en. In: (), p. 13.

[30] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. "Feature Pyramid Networks for Object Detection". In: *arXiv:1612.03144 [cs]* (Apr. 2017). arXiv: 1612.03144. URL: http://arxiv.org/abs/1612.03144 (visited on 04/13/2020).

[31] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. "Focal Loss for Dense Object Detection". In: *arXiv:1708.02002 [cs]* (Feb. 2018). arXiv: 1708.02002. URL: http://arxiv.org/abs/1708.02002 (visited on 04/07/2020).

[32] 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". In: *arXiv:1405.0312 [cs]* (Feb. 2015). arXiv: 1405.0312. URL: http://arxiv.org/abs/1405.0312 (visited on 08/19/2020).

[33] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. "SSD: Single Shot MultiBox Detector". In: *arXiv:1512.02325 [cs]* 9905 (2016). arXiv: 1512.02325, pp. 21–37. DOI: 10.1007/978-3-319-46448-0_2. URL: http://arxiv.org/abs/1512.02325 (visited on 08/19/2020).

[34] D. Lowe. "Object recognition from local scale-invariant features". en. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Kerkyra, Greece: IEEE, 1999, 1150–1157 vol.2. ISBN: 978-0-7695-0164-2. DOI: 10.1109/ICCV.1999.790410. URL: http://ieeexplore.ieee.org/document/790410/ (visited on 08/18/2020).

[35] D. G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". en. In: *International Journal of Computer Vision* 60.2 (Nov. 2004), pp. 91–110. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000029664.99615.94. URL: http://link.springer.com/10.1023/B:VISI.0000029664.99615.94 (visited on 08/19/2020).

[36] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Y. Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[37] *MEDLINE, PubMed, and PMC (PubMed Central): How are they different?* en. 2019. URL: https://www.nlm.nih.gov/bsd/difference.html (visited on 08/19/2020).

[38] K. Mikolajczyk and C. Schmid. "Scale & Affine Invariant Interest Point Detectors". en. In: *International Journal of Computer Vision* 60.1 (Oct. 2004), pp. 63–86. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000027790.02288.f2. URL: http://link.springer.com/10.1023/B:VISI.0000027790.02288.f2 (visited on 08/19/2020).

[39] H. Müller, V. Andrearczyk, O. J. del Toro, A. Dhrangadhariya, R. Schaer, and M. Atzori. "Studying Public Medical Images from the Open Access Literature and Social Networks for Model Training and Knowledge Extraction". In: *International Conference on Multimedia Modeling*. Springer. 2020, pp. 553–564.

[40] H. Müller, J. Kalpathy-Cramer, C. E. Kahn Jr, and W. Hersh. "Comparing the quality of accessing medical literature using content-based visual and textual information retrieval". In: *Medical Imaging 2009: Advanced PACS-based Imaging Informatics and Therapeutic Applications*. Vol. 7264. International Society for Optics and Photonics. 2009, p. 726405.

[41] H. Müller, F. Meriaudeau, A. Foncubierta-Rodríguez, D. Markonis, and A. Chhatkuli. "Separating compound figures in journal articles to allow for subfigure classification". In: SPIE, Medical Imaging (2013).

[42] H. Müller, N. Michoux, D. Bandon, and A. Geissbuhler. "A review of content-based image retrieval systems in medical applications—clinical benefits and future directions". In: *International journal of medical informatics* 73.1 (2004), pp. 1–23.

[43] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, and J. Walsh. "Deep Learning vs. Traditional Computer Vision". en. In: *Advances in Computer Vision*. Ed. by K. Arai and S. Kapoor. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2020, pp. 128–144. ISBN: 978-3-030-17795-9. DOI: 10.1007/978-3-030-17795-9_10.

[44] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019,

pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[45] A. Rahimi and B. Recht. "Random Features for Large-Scale Kernel Machines". In: *Advances in Neural Information Processing Systems 20*. Ed. by J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis. Curran Associates, Inc., 2008, pp. 1177–1184. URL: http://papers.nips.cc/paper/3182-random-features-for-large-scale-kernel-machines.pdf (visited on 08/20/2020).

[46] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. "You Only Look Once: Unified, Real-Time Object Detection". In: *arXiv:1506.02640 [cs]* (May 2016). arXiv: 1506.02640. URL: http://arxiv.org/abs/1506.02640 (visited on 08/19/2020).

[47] J. Redmon and A. Farhadi. "YOLO9000: Better, Faster, Stronger". In: *arxiv:1612.08242 [cs]* (Dec. 2016). arxiv: 1612.08242. URL: http://arxiv.org/abs/1612.08242 (visited on 08/19/2020).

[48] J. Redmon and A. Farhadi. "YOLOv3: An Incremental Improvement". In: *arXiv:1804.02767 [cs]* (Apr. 2018). arXiv: 1804.02767. URL: http://arxiv.org/abs/1804.02767 (visited on 08/19/2020).

[49] S. Ren, K. He, R. Girshick, and J. Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *arXiv:1506.01497 [cs]* (Jan. 2016). arXiv: 1506.01497. URL: http://arxiv.org/abs/1506.01497 (visited on 04/13/2020).

[50] A. Rosebrock. *Pedestrian detection OpenCV*. en. 2015. URL: https://www.pyimagesearch.com/2015/11/09/pedestrian-detection-opencv/ (visited on 08/22/2020).

[51] 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". In: *arXiv:1409.0575 [cs]* (Jan. 2015). arXiv: 1409.0575. URL: http://arxiv.org/abs/1409.0575 (visited on 08/19/2020).

[52] M. Saifee. *Pytorch vs Tensorflow in 2020*. en. 2020. URL: https://towardsdatascience.com/pytorch-vs-tensorflow-in-2020-fe237862fae1 (visited on 08/19/2020).

[53] H. Shatkay, N. Chen, and D. Blostein. "Integrating image data into biomedical text categorization". In: *Bioinformatics* 22.14 (2006), e446–e453.

[54] A. Suleiman and V. Sze. "Energy-Efficient HOG-based Object Detection at 1080HD 60 fps with Multi-Scale Support". In: Oct. 2014. DOI: 10.13140/2.1.4752.0964.

[55] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions". en. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE, June 2015, pp. 1–9. ISBN: 978-1-4673-6964-0. DOI: 10.1109/CVPR.2015.7298594. URL: http://ieeexplore.ieee.org/document/7298594/ (visited on 08/19/2020).

[56] M. Tan, R. Pang, and Q. V. Le. "EfficientDet: Scalable and Efficient Object Detection". In: *arXiv:1911.09070 [cs, eess]* (Apr. 2020). arXiv: 1911.09070. URL: http://arxiv.org/abs/1911.09070 (visited on 04/16/2020).

[57] E. Tola, V. Lepetit, and P. Fua. "Daisy: An Efficient Dense Descriptor Applied to Wide Baseline Stereo". In: *IEEE transactions on pattern analysis and machine intelligence* 32 (May 2010), pp. 815–30. DOI: 10.1109/TPAMI.2009.77.

[58] S. Tsutsui and D. Crandall. "A Data Driven Approach for Compound Figure Separation Using Convolutional Neural Networks". In: *arXiv:1703.05105 [cs]* (Aug. 2017). arXiv: 1703.05105. URL: http://arxiv.org/abs/1703.05105 (visited on 04/01/2020).

[59] P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features". en. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 1. Kauai, HI, USA: IEEE Comput. Soc, 2001, pp. I–511–I–518. ISBN: 978-0-7695-1272-3. DOI: 10.1109/CVPR.2001.990517. URL: http://ieeexplore.ieee.org/document/990517/ (visited on 08/19/2020).

[60] Z. Wang, B. Fan, and F. Wu. "Local Intensity Order Pattern for feature description". In: *2011 International Conference on Computer Vision* (2011). DOI: 10.1109/ICCV.2011.6126294.

[61] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. *Detectron2*. https://github.com/facebookresearch/detectron2. 2019.

[62] D. You, S. Antani, D. Demner-Fushman, V. Govindaraju, and G. Thoma. "Detecting Figure-Panel Labels in Medical Journal Articles Using MRF". In: tex.ids: DetectingFigurePanelLabels. Oct. 2011, pp. 967–971. DOI: 10.1109/ICDAR.2011.196.

[63] X. Zhou, D. Wang, and P. Krähenbühl. "Objects as Points". In: *arXiv:1904.07850 [cs]* (Apr. 2019). arXiv: 1904.07850. URL: http://arxiv.org/abs/1904.07850 (visited on 04/13/2020).

[64] J. Zou, S. Antani, and G. Thoma. "Localizing and Recognizing Labels for Multi-Panel Figures in Biomedical Journals". In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. Vol. 01. ISSN: 2379-2140. Nov. 2017, pp. 753–758. DOI: 10.1109/ICDAR.2017.128.

[65] J. Zou, G. Thoma, and S. Antani. "Unified Deep Neural Network for Segmentation and Labeling of Multipanel Biomedical Figures". en. In: *Journal of the Association for Information Science and Technology* (2019). _eprint: https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/asi.24334. ISSN: 2330-1643. DOI: 10.1002/asi.24334. URL: https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/asi.24334 (visited on 04/01/2020).