Toward Automatic Vector Extraction from Scanned Historical Aerial Photo Indexes

Salim Malek, Elisa Mariarosaria Farella, Giulio Perda, Gianluca Cantoro, and Fabio Remondino

Abstract

National archives worldwide have recently dedicated significant efforts to preserve fragile historical collections (e.g., maps, documents, photographs) through scanning and digitization. Advancements in digital technologies have opened new opportunities for processing and accessing these materials; however, as these collections grow, significant challenges arise in efficiently managing, linking, and using vast volumes of digital data. This paper addresses the challenge imposed by the digital transformation of vast archive collections, with a focus on scanned materials preserved in historical photographic and mapping archives and intended for the creation of digital databases and online catalogues. This study targets the semi-automatic vectorization of aerial photo indexes (APIs), often referred to as finding aids for aerial reconnaissance sorties (i.e., maps on which aerial photo footprints [each corresponding to a historical aerial photo] from different surveying flight missions were manually drawn onto base topographic maps). Aerial images were captured during the 20th century for military, reconnaissance, and mapping purposes, and their footprints were then manually transposed onto reference maps. Currently, archives and mapping institutions managing creating digital databases and catalogues are manually addressing a demanding and time-consuming vectorization task, and automated solutions are highly needed. This contribution proposes a novel and comprehensive semi-automatic vectorization pipeline that leverages and integrates computer vision techniques and uses deep learning and computational geometry for detecting, isolating, and polygonizing aerial image footprints in APIs. The methodology is applied to a large collection of APIs scanned by the Italian National Historical AirPhoto Archive at the Italian National Institute for Cataloguing and Documentation of the Ministry of Culture and will be added to the online database and WebGIS catalogue recently set up to disseminate copies of historical aerial photographs.

Introduction

Over the last few decades, worldwide national cartographic and photographic archives have been increasingly engaged in scanning and digitizing their historical analog collections of photos and maps to preserve the content of these fragile materials from physical degradation and to disseminate them to a wider public. Advancements in digital data processing have created new opportunities for using, linking, and accessing these archival sources.

Three main directions can be easily identified when analyzing recent research exploiting these historical sources: evaluating the quality of digitized data for diverse two- and three-dimensional processing tasks, developing methods for the accurate georeferencing of historical data, and advancing techniques for the automatic extraction and vectorization of map features. Several studies have examined the quality of digitized historical aerial images for mapping purposes, with a focus on radiometric and geometric accuracy and overall fidelity of these sources, as well as the assessment of three-dimensional derived products of different scanning solutions (Armenakis *et al.* 2003; Tuno *et al.* 2022; Kostrzewa *et al.* 2024). Further works provide guidelines and standards for digitization of heterogeneous historical imaging (Malaperdas 2021).

Georeferencing of historical digitized sources preceding vectorization ensures that spatial data are aligned with real-world coordinates and that a consistent relationship between features is preserved. In turn, georeferenced vector data supports broad geospatial workflows and applications—including the creation of spatially indexed online catalogues, change detection, and three-dimensional modeling—while also preserving historical spatial information in a reusable digital format (Poli *et al.* 2020; Craciun and Le Bris 2022; Farella *et al.* 2022). In the past few decades, artificial neural networks and several image-matching strategies and tools have been proposed to improve the accuracy and automation for the spatial alignment of historical and modern maps and aerial imagery for georeferencing (Yilmaz and Gullu 2012; Cantoro 2014, 2017; Luft and Schiewe 2021; Zhang *et al.* 2022).

Vectorization is the process of converting raster imagery into structured numerical formats. It plays a central role in enhancing the accessibility and research potential of historical data across multiple disciplines by connecting historical analog archives with modern analytical methods. This transformation, especially in historical topographic maps, involves extracting discrete geographic features and representing them as points, lines, or polygons within a geographic information system (GIS). Solutions implemented for increasing automation in the vectorization of historical maps focus on extracting specific and recurrent features of the built and natural environments (e.g., buildings, streets, vegetation; texts and labels visible in cadastral and topographic city-scale maps; and various landscape features and manually drawn information from larger scale maps (DeKruger and Hunt 1994; Hancer and Samet 2011; Iosifescu *et al.* 2016; Gede *et al.* 2020; Le Riche 2020; Chen *et al.* 2021; Schlegel 2021; Dunesme *et al.* 2022; Petitpierre and Guhennec 2023; Chen *et al.* 2024).

Semi-automated approaches, including edge detection, optical character recognition (OCR), and machine-learning algorithms, offer the acceleration of feature extraction for extensive data sets, but the complete automatic vectorization of scanned maps is far from final and remains an error-prone process due to a confluence of technical challenges (Steven *et al.* 2010; Szendrei *et al.* 2011; Arteaga 2013; Chiang *et al.* 2013; Chrysovalantis and Nikolaos 2020; Gede *et al.* 2020; Martínek *et al.* 2020; Chen *et al.* 2024). Many of these challenges originate from limitations in the input data, such as image degradation and noise that complicate initial feature detection, edge detection, and line tracing. Subsequent stages are hindered by difficulties in maintaining geometric accuracy during generalization and in performing correct semantic labeling due to archaic symbology. Moreover, deep learning (DL) models experience difficulty with heterogeneous map styles, limiting the full automation of the workflow.

Because of the current limitations in achieving full automation and the absence of a one-size-fits-all solution capable of extracting all

Salim Malek, Elisa Mariarosaria Farella, Giulio Perda, and Fabio Remondino are with the 3D Optical Metrology (3DOM) Unit, Bruno Kessler Foundation (FBK), Trento, Italy (remondino@fbk.eu).

Gianluca Cantoro is with ISPC-CNR, Italy.

Corresponding author: Fabio Remondino (remondino@fbk.eu)

Received August 25, 2025, accepted December 24, 2025.

Photogrammetric Engineering & Remote Sensing
Vol. 92, No. X, MONTH 2026, pp. 1–xx.
0099-1112/22/1–xx

doi: 10.14358/PERS.25-00133R3

relevant features for different applications and research needs, flexible approaches are required. Overcoming these challenges demands hybrid strategies that integrate image preprocessing, adaptable semi-automated vectorization methods to improve generalization, and postprocessing validation to ensure topological consistency.

Within this context and with a focus on improving the efforts for the digitization of historical aerial photo indexes (APIs) undertaken by various national photographic archives, this paper addresses two significant challenges of digital transformation, georeferencing and vectorization of vast collections of APIs for creating accessible and explorable online databases and catalogues (FOTOTECA, 2026) (Figure 1; Archivio Aerofototeca Nazionale 2026; Bundesamt für Eich- und Vermessungswesen 2026; Czechoslovakia Government, Archives 2026; Department of Cadastre and Surveying 2026; Federal Office of Topography, Swisstopo 2026; National Collection of Aerial Photography 2026; Norwegian Mapping Authority 2026; République Française, National Institute of Geographic and Forest Information 2026; Schweizerische Eidgenossenschaft 2026).

APIs (commonly called finding aid for aerial reconnaissance sorties) are related to historical aerial photographs taken from the early 20th century onward, collected primarily for military and mapping purposes. The APIs contain footprints and metadata of flight paths and photographs acquired during several surveying campaigns (Figure 1). Each API contains one or more flight sorties that are represented as sequences of consecutive overlapping polygons with manually drawn information (e.g., sortie number, survey's company name or military authority, print and negative number, flight direction) beside the traditional flight metadata (i.e., camera focal length, type of camera,

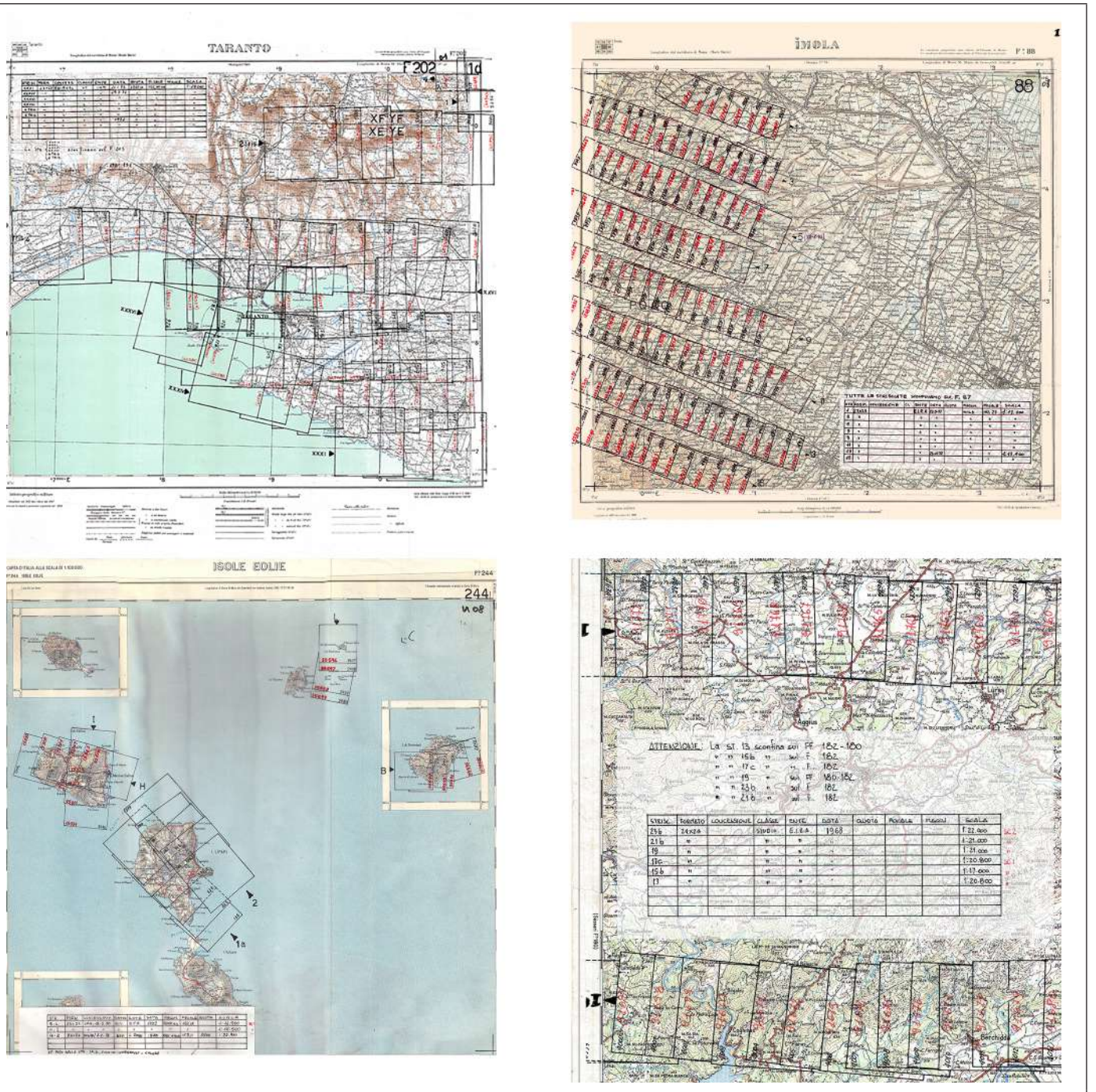

Figure 1. (Top) Complete and (bottom) magnified aerial photo indexes featuring aerial image footprints (i.e., drawn polygons) and flight metadata (tables) to be vectorized.

flying height, date of flight, image scale). The APIs were manually drawn by different operators onto topographic maps (those considered in this work were produced by the Italian Military Geographic Institute (IGM) at a national scale of 1:100 000). Scanned APIs features various geometric resolutions up to 8000×8000 pixels.

This paper reports a semi-automatic pipeline assembled for the georeferencing (*APIs Georeferencing*) and vectorization (*APIs Vectorization*) of more than 4000 APIs scanned by the Italian photographic archive (Aerofototeca Nazionale). This vectorization framework supports the process of enrichment of the digital historical catalogues of Aerofototeca Nazionale, which has launched a web portal hosting some series of georeferenced and vectorized APIs (Archivio Aerofototeca Nazionale 2026).

This work principally contributes a comprehensive vectorization workflow that integrates techniques from computer vision, DL, and computational geometry to ensure both robustness and efficiency in the vectorization process, as well as generalizability across the country and throughout decades (APIs reference period, 1930–1990; *APIs Vectorization*). A cornerstone of this work is a novel segmentation model, which refines the DeepLab architecture to specifically address the challenges of background removal and severe class imbalance, thereby enabling highly accurate feature extraction from complex historical imagery (*Background removal*; Chen *et al.* 2018).

The proposed approach demonstrates promising performance, achieving high intersection over union (IoU), sensitivity, and precision (*Results and Discussion*). The methodology shows strong potential for application to similar data sets worldwide with minimal adaptation.

Related Work

The vectorization of (historical) images and maps has been extensively studied in the fields of computer vision, GIS, and remote sensing. Previous research has explored various techniques, ranging from traditional image-processing methods to modern learning-based approaches. To the best of the authors' knowledge, no previous work has addressed the vectorization problem examined in this paper, although some methodologies (based on morphological analyses and DL techniques, which are both used in this study) have been used for solving similar issues, with a focus primarily on road vectorization or other urban features.

Early Vectorization Approaches

Early methods relied on manual vectorization by cartographers who traced raster images to extract vector features (Douglas and Peucker 1973). Although accurate, this process is labor intensive, prompting the development of semi-automated techniques profiting from the development of useful computer vision techniques such as skeletonization, line detection, morphological operations, and edge detection (Ballard and Brown 1982; Gonzalez and Woods 2008).

Among the first notable contributions to map vectorization using computer vision techniques, Hori and Okazaki (1992) proposed a generic model that formalizes the problem as an optimization task, where the determination of optimal building shapes is achieved through a mathematical cost function. However, skeletonization results provided many artifacts and deformations of the shapes. To address these challenges, Janssen and Vossepoel (1997) introduced a postprocessing stage applied to the results by incorporating geometric regularization rules and adaptive thresholds. This refinement process filters nodes and segments while optimizing angular relationships, thereby enhancing the topological quality of the vectorized output. Nevertheless, the method is limited by the dependence on the initial global thickness, with risks of over-simplification in the postprocessing phase, high computational costs, and reduced support for different drawing conventions.

Itonaga *et al.* (2003) proposed a new approach for the automatic extraction of road networks from maps. First, the road area is extracted from the map image by stochastic relaxation based on the geometrical properties of the road area and the background area. Then, thinning and piecewise-linear approximation are applied to the road area to extract the road network information, which contains the graphic structure of nodes and links. In this process, the distortion near junctions produced by thinning is corrected, and the nodes are optimally rearranged so that the road network represents the centers of the roads. However, this method has difficulty in areas where road features are intersected with overlapping symbols or labels; the quality of the results is tied to the setting of parameter values and is especially map specific. In Chiang and Knoblock (2008, 2009), road pixels are first skeletonized to produce a thinned raster representation, then the template-based intersection-detection algorithm locates and extracts intersection geometries, and finally, the road vectors are reconstructed via pixel tracing along the skeletonized network. The method is not fully automatic and requires some user-label inputs, and the pixel extraction quality is conditioned by the scan and color quality, as well as the presence of overlapping features.

Henderson and Linton (2009) proposed a novel framework of map feature extraction; the methodology applies a color histogram-based classification derived from cartographic color conventions, followed by a tensor voting-based segmentation for linear feature detection, including roads, rivers, and intersections. To improve the accuracy of image-processing outputs, researchers have also incorporated machine-learning techniques for the segmentation and classification of topographic maps, such as Fuzzy C-Means clustering algorithm, artificial neural networks, and super vector machine algorithms (Chiang and Knoblock 2006; Xie *et al.* 2006; Karabork *et al.* 2008; Miao *et al.* 2015). However, dependence on color histograms causes this method to be vulnerable to scanning artifacts and noise, and the variation across different map series may limit its generalization.

These classical computer vision and initial rule-based methods face difficulties with APIs because they require predefined symbols and shapes, consistent colors, or sharp edges.

DL Vectorization Approaches

DL has emerged as a powerful paradigm within machine learning, demonstrating superior performance in complex problem domains, particularly in tasks involving unstructured data such as images (LeCun *et al.* 2015; Hatcher and Yu 2018). In map vectorization, DL techniques are primarily used to classify and segment objects of interest (e.g., roads, buildings, water bodies) from raster maps and satellite imagery. Zhao *et al.* (2022) proposed a method to extract building blocks from historical maps based on deep object attention networks, using the OCRNet framework with multiple attention mechanisms to improve the results. Schlegel (2023) applied a combination of object-based image analysis and vectorization methods on historical maps to extract buildings and roads for multi-temporal analyses. Chen *et al.* (2024) focused on shape vectorization from large-scale historical maps, especially city maps; the authors designed a supervised DL pipeline to reliably extract closed shapes from historical maps, working specifically with Paris atlases from the 19th and early 20th centuries. Sertel *et al.* (2024) explored transformer-based segmentation methods for geospatial object extraction from historical maps, showing their superior performance over traditional convolutional neural network (CNN)-based architectures. Jiao *et al.* (2024) proposed a novel framework to vectorize and classify roads from historical maps as a cutting-edge approach in automated road extraction using DL techniques. Jiao *et al.* (2022) also proposed an efficient methodology to automatically generate training data through symbol reconstruction for road extraction.

DL is particularly advantageous for the vectorization of APIs because APIs present various challenges that DL networks are designed to solve, including complex patterns, variable drawing styles and line thickness, noisy data, and overlapping items. DL can directly learn features from the given data, increasing generalization and robustness (e.g., to styles or scanning artefacts).

APIs Georeferencing

Georeferencing plays a crucial role in the vectorization pipeline and in the digital transformation of archival cartographic data. This unavoidable process ensures that APIs and aerial image footprints are spatially aligned with actual geographic reference systems and within webGIS and geo-portal infrastructures. Most of the historical Italian APIs were hand drawn on copies of IGM map sheets at 1:100 000 scale (277

Figure 2. Available georeferenced Italian Military Geographic Institute sheet (left) and an aerial photo index over the same area to be georeferenced.

sheets covering the entire country) or other cartographic basemaps. Leveraging these basemaps, thousands of APIs were produced and later digitized and categorized in different series or collections. The original georeferenced cartographic sheets provided by the IGM served in this work as a spatial reference framework for the APIs because they match the actual national cartographic basemaps.

Considering the significant number of APIs to be georeferenced and then vectorized (more than 4000), the AutoGR-Toolkit is used (Cantoro 2014). This tool integrates several Python scripts and libraries that are designed to search for subpixel precision correspondences between images to be aligned or georeferenced. In the current version of the AutoGR-Toolkit, handcrafted methods such as SIFT, SURF, or ORB are coupled with new learning-based methods such as SuperGlue (Lowe 2004; Bay *et al.* 2006; Rublee *et al.* 2011; Sarlin *et al.* 2020). For the available APIs, handcrafted methods ensured accurate results in most cases, whereas learning-based approaches were used in cases of exceptionally different basemaps (in cartographic features or scale; Figure 2).

Considering the 1:100 000 scale of the IGM maps used for the APIs, the large number of automatically extracted tie points enabled consistent and reliable georeferencing, achieving an average planimetric error of approximately 5 to 10 m across the entire data set, despite the variability of the maps and the use of both handcrafted and learning-based approaches.

APIs Vectorization

Following the georeferencing of individual APIs (*APIs Georeferencing*), the vectorization procedure comprises several stages (Figure 3). First, the background is removed (*Background removal*), then lines defining the footprint polygons are identified (*Line Detection*), arrows that indicate the various flight directions of each strip are then detected (*Arrow Detection*), and finally, individual footprint polygons are created (*Polygon Creation*). Compared with existing vectorization solutions (*Related Work*), often relying only on morphological analyses or DL semantic segmentation approaches, the proposed pipeline efficiently integrates segmentation with geometric reasoning to create structured and semantically coherent polygonal sequences. The automatic detection of arrow markers serves as geometric anchors to initialize the identification of each strip, introducing directional and spatial constraints. Class imbalances and thin object detection are improved by proposing a refined DeepLabV3 architecture (RefDeepLab), which preserves crucial geometric details in large imagery during the upscaling. Furthermore, compared with many state-of-the-art approaches frequently experiencing limited generalization, the implemented pipeline (tested and applied on more than 4000 heterogeneous APIs from different map series) proved its robustness to different times, styles, and scanning conditions.

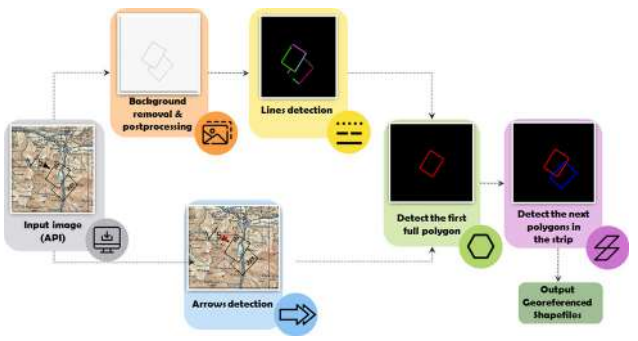

Figure 3. Implemented vectorization workflow. API = aerial photo index.

Background removal

To allow efficient identification and extraction of objects of interest (strips, polygons), the cartographic background must be removed (Figure 4). The DeepLabV3 model is adapted to a binary segmentation system to allow polygonal features extraction (specifically, rectangles) while filtering out background elements (Chen *et al.* 2018). DeepLabV3 offers advancement over the original DeepLab architecture, delivering improved performance for this task. The adopted model uses ResNet101, pretrained on the ImageNet data set, as its backbone architecture (He *et al.* 2016). The core innovation of DeepLabV3 lies in its use of the Atrous Spatial Pyramid Pooling module (Yu and Koltun 2016). The atrous convolutions are based on expanding the window size without increasing the number of the weights, which allows the stride to remain constant but with a larger field of view without increasing the number of parameters or the amount of computation (Figure 5).

Mathematically, let F: $\mathbb{Z}^2 \rightarrow \mathbb{R}$ be a discrete function (in this case, the image). Let $\Omega r = [-r, r]^2 \cap \mathbb{Z}^2$ and let k: $\Omega r \rightarrow \mathbb{R}$ be a discrete filter of size $(2r + 1)^2$. The discrete convolution operator "*" can be defined as Equation 1 (Yu and Koltun 2016):

$$(F * k)(p) = \sum_{s+t=p} F(s)k(t) \quad (1)$$

where $\mathbb{Z}$ denotes the set of integers, $\mathbb{R}$ is the set of real numbers, and r is a strictly positive integer.

Equation 1 can be generalized by introducing the dilatation factor l. The new discrete convolution operator $*_l$ can be defined as Equation 2:

$$(F *_l k)(p) = \sum_{s+lt=p}^{F} (s)k(t) \quad (2)$$

The standard DeepLab architecture uses a shallow decoder that performs a couple of upsampling operations by a factor of 4. This design contributes to the model's computational efficiency and rapid training, but it can lead to the loss of fine-grained details, which is a significant drawback when segmenting imbalanced data sets where minority classes are often characterized by small features. To address this limitation, this study proposes a refined architecture, RefDeepLab. This model replaces the two upsampling steps with a progressive decoder that upsamples feature maps by a factor of 2 at each stage.

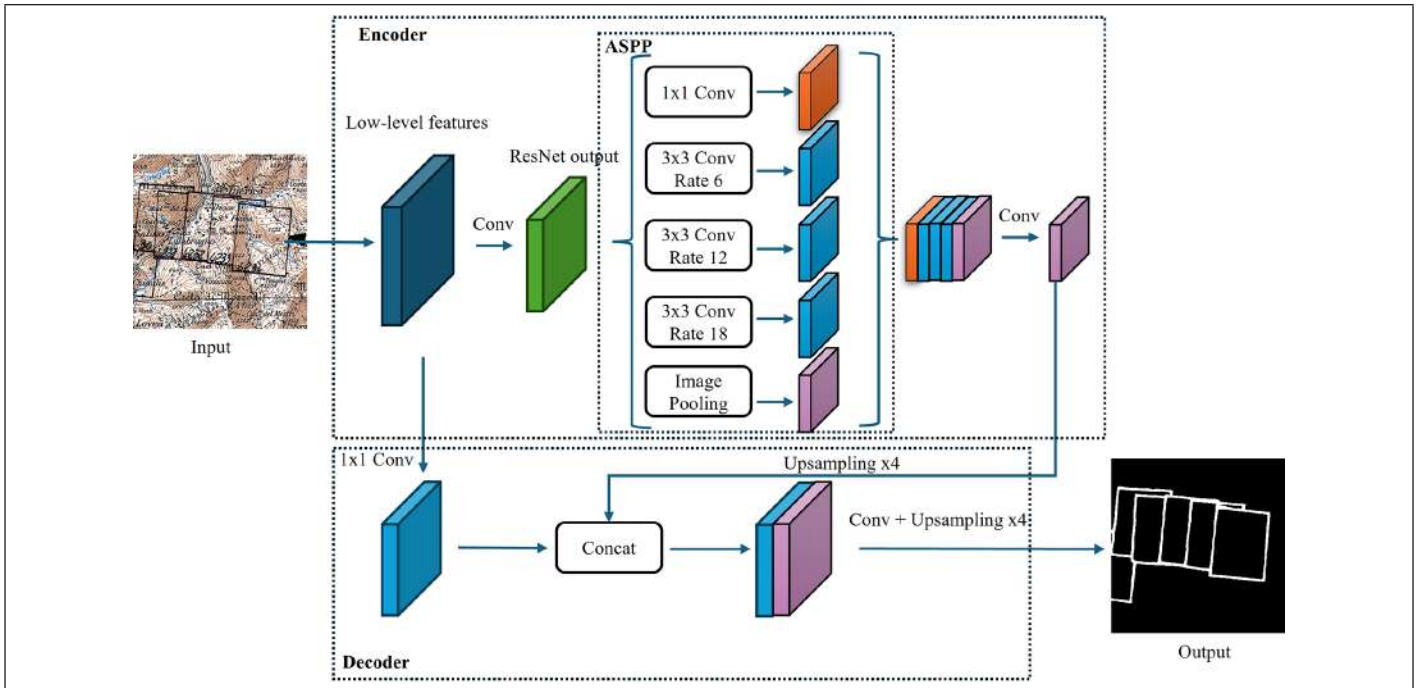

Figure 4. Network structure of the background removal model based on DeepLabV3. ASPP = Atrous Spatial Pyramid Pooling.

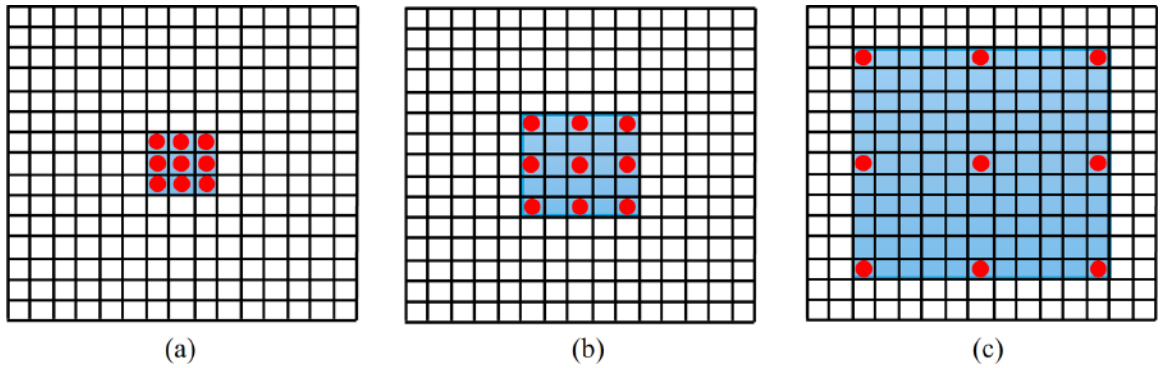

Figure 5. Principle of dilated convolution. (a) $l = 1$: Each output element has a receptive field of 3×3. (b) $l = 2$: Each output element has a receptive field of 5×5. (c) $l = 5$: Each output element has a receptive field of 11×11.

Each upsampling step is followed by a refining block that incorporates a skip connection from the corresponding encoder layer, followed by an Atrous Spatial Pyramid Pooling block. The complete architecture of RefDeepLab is illustrated in Figure 6.

The output generated by the RefDeepLab model necessitates post-processing to enhance its quality. A skeletonization technique is used to reduce polygon (line) thickness to single-pixel width, thereby facilitating subsequent processing stages (Zhang and Suen 1984; Maragos and Schafer 1986; Serra 1988).

Line Detection

Line (or segment) detection is performed using the Hough transform (Hough 1959). As a parametric feature detection method, the Hough transform enables the isolation of predefined shapes within images. The classical approach operates most effectively on regular curves with known parametric equations, making it particularly suitable for line, circle, and ellipse detection in computer vision applications (Jain *et al.* 1995). A line segment in a Cartesian coordinate system Oxy can analytically be defined as follows (Equation 3):

$$y = ax + b \tag{3}$$

In the polar coordinate system (r, θ), it is described in Equation 4:

$$r = x.\cos\theta + y.\sin\theta \tag{4}$$

where r is the length of a normal from the origin to this line and θ is the orientation of r with respect to the x axis.

If for a given point (x_0, y_0) the family of lines that passes through it is plotted, this study obtains a sinusoid in the polar coordinate system (r, θ).

Given a set of collinear points in a Cartesian coordinate system, the curves in the polar coordinate system intersect at a common point. This intersection phenomenon forms the theoretical foundation of the Hough transformation for line detection, where the maximal curve intersections in the polar Hough parameter space identify the most probable lines in image space.

The result of this step is a list of segments s_i. Each segment is defined by its two boundary points $p_{i,1}, p_{i,2}$, where $p_{i,j} = (x_{i,j}, y_{i,j})$ for $j \in \{1, 2\}$.

Arrow Detection

The automated processing pipeline simultaneously performs line detection and arrow marker identification, where the arrows, accompanied by alphanumeric codes, serve as critical reference indicators for the strips within each API. This arrow-detection phase provides essential spatial reference points that initiate the subsequent polygon extraction process for individual strips.

The arrow-detection module uses YOLOv8 (Bochkovskiy *et al.* 2020); the module incorporates a recent iteration of the YOLO (you only look once) architecture originally developed by Redmon *et al.* (2016). As a single-stage object detector, YOLOv8 uses a CNN architecture in which the initial 24 convolutional layers perform

hierarchical feature extraction, and the two fully subsequent connected layers simultaneously predict both class probabilities and bounding box coordinates through regression (Redmon *et al.* 2016). The convolutional layers are trained on an ImageNet data set for classification, using the input data at half resolution (224×224) and doubling them in the detection phase (Deng *et al.* 2009). The architecture of YOLO has undergone continuous refinement, maintaining its position in different applications as a state-of-the-art solution for real-time object detection tasks such as medicine, forest management and monitoring, remote sensing, agriculture, and industrial manufacturing and quality control (Bochkovskiy *et al.* 2020; Parico and Ahamed 2020; Ikechukwu and Akin 2022; Aman *et al.* 2023; Liu *et al.* 2024; Zuo *et al.* 2024; Akdoğan *et al.* 2025; Chen *et al.* 2025; Deng *et al.* 2025; Jin *et al.* 2025; Kang *et al.* 2025; Mehta *et al.* 2025; Sun *et al.* 2025; Wang, Han *et al.* 2025; Wang, Lin *et al.* 2025; Zhao *et al.* 2025; Zhou *et al.* 2025).

Polygon Creation

The line and arrow positions, as determined in the preceding steps, serve as the basis for polygon identification. The arrows facilitate the localization of the initial complete polygon within each corresponding strip. After the dimensions of this polygon are established, they can be systematically applied to reconstruct the remaining polygons in the strip, accounting for their overlapping configurations.

Detection of the First Polygon

The accurate detection of the first, non-overlapped polygon is critical. This principle is based on the operational assumption that each strip corresponds to a single flight mission with fixed altitude and camera parameters (reported in the metadata of each API), which should result in uniform polygon dimensions across the strip. However, inconsistencies in the manual drawing of strips on the original map sheets, terrain variations, or deformations of the scanned map introduce occasional size variations, which violate this assumption. Nevertheless, considering the purpose of the polygon vectorization, the assumption can be accepted. Consider a strip constituted by a set of N polygons $\{P_i \mid i = 1, \ldots, N\}$. Each polygon P_i is formed by four boundary segments, denoted $s_{i,j}$ for $j = 1, 2, 3$, and 4. The detection of the initial polygon (P_1) begins by computing the distances between the arrow and all available segments. The segment located nearest to the arrow is identified as the first segment of P_1 ($s_{1,1}$). Additionally, the directional vector from the arrow to $s_{1,1}$ (D_1) is recorded for subsequent processing. Figure 7 illustrates the selection of the first segment in different cases.

The subsequent step involves identifying the two segments perpendicular to $s_{1,1}$ ($s_{1,2}$ and $s_{1,4}$). This is achieved by computing the angles between $s_{1,1}$ and all other segments, retaining only those segments that form an angle of approximately 90°. To ensure accurate selection and minimize false detections, the following criteria are applied:

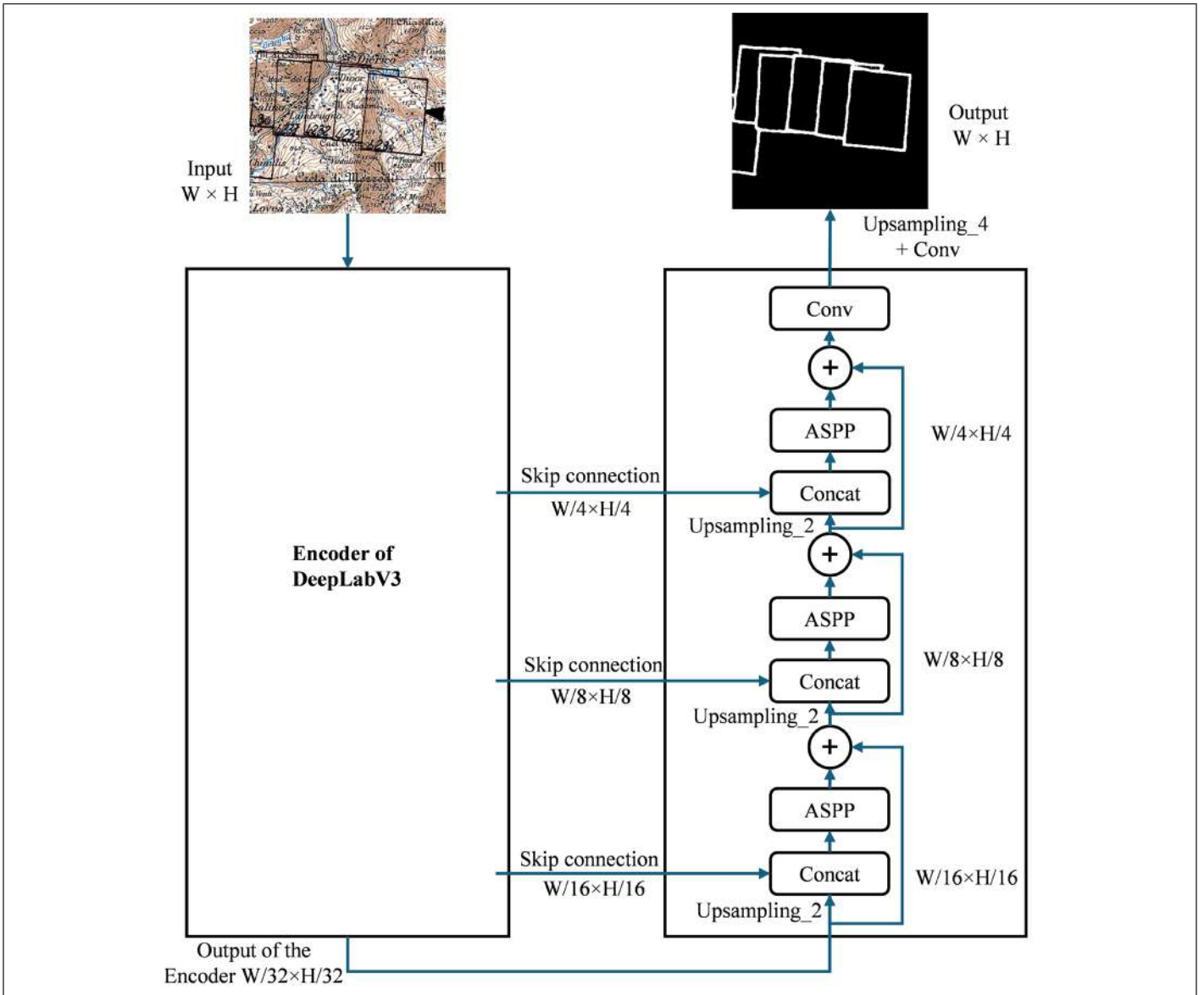

Figure 6. Architecture of the proposed RefDeepLab model. ASPP = Atrous Spatial Pyramid Pooling.

- Positional constraint: The reference segment $s_{1,1}$ must lie spatially between $s_{1,2}$ and $s_{1,4}$.
- Proximity priority: In cases where multiple valid candidates satisfy the angular condition, the segment closest to $s_{1,1}$ is selected.

The final step involves detecting the last segment $s_{1,2}$. This segment is identified as being parallel to $s_{1,1}$ and situated between the two previously detected perpendicular segments $s_{1,2}$ and $s_{1,4}$ while maintaining the same directional orientation as D_1. On detection, the first polygon (rectangle) is fully delineated, allowing for the extraction of its dimensions (length and width; Figure 8). The vertices of the polygon are determined by computing the intersections between the constituent segments. These corner points are then stored in an ordered list, thereby defining the polygon by its four vertices.

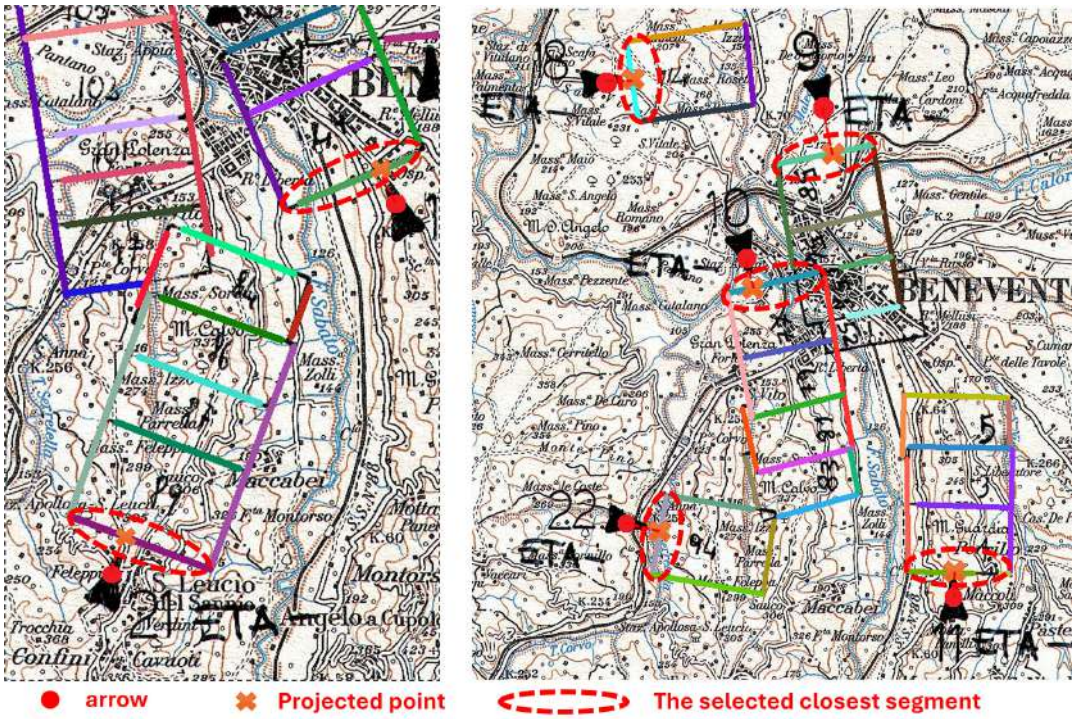

Figure 7. Detection of the closest segments to arrows.

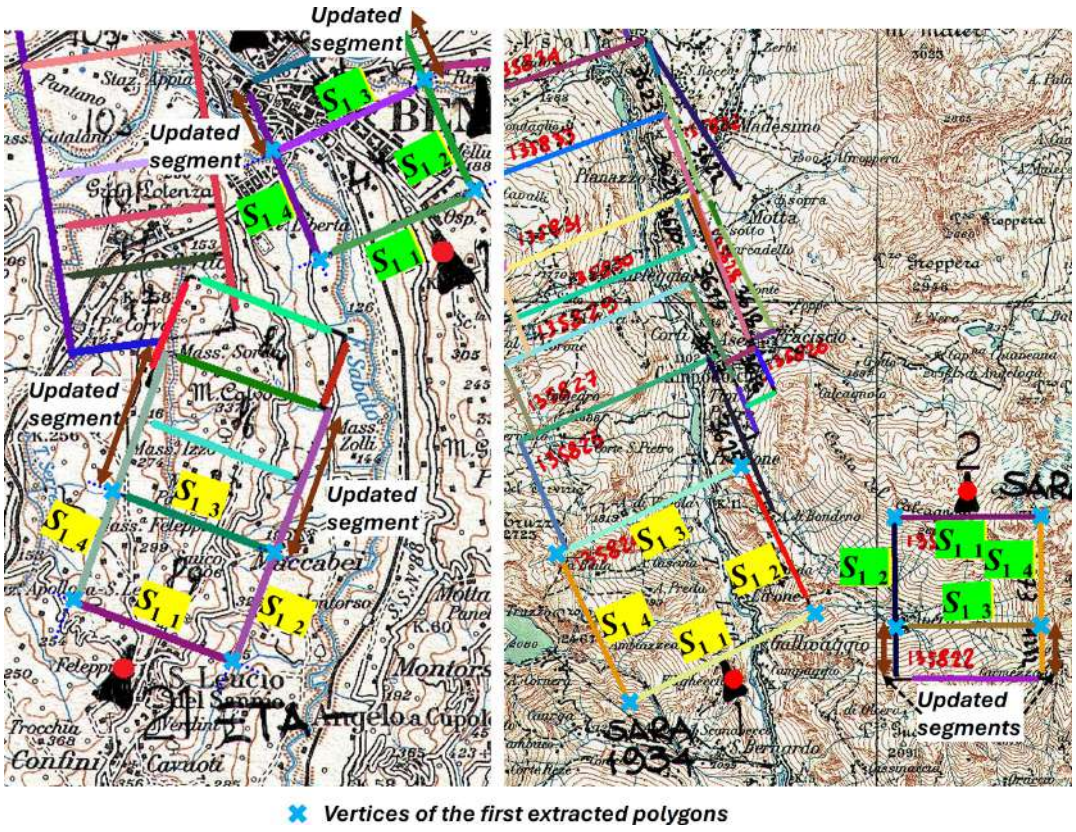

Figure 8. Detection of the first polygon P_1.

Automated Identification of Subsequent Polygons

The detection of subsequent polygons (P_i, where $i = 2{:}N$ and N represents the total number of polygons on the current strip) follows an iterative procedure. The algorithm is initiated by taking the segment $s_{i-1,3}$ from the previously detected polygon P_{i-1} to identify the corresponding segment $s_{i,3}$ of the current polygon P_i. This search is conducted along the strip's primary direction (D_1) to locate the next parallel segment.

Subsequently, the two perpendicular segments ($s_{i,2}$ and $s_{i,4}$) are detected, adhering to the same geometric constraints outlined in the initial polygon detection phase. Furthermore, a safety margin of 25% is incorporated into the searching criterion. Consequently, for the two perpendicular segments $s_{i,2}$ and $s_{i,4}$ to be considered parallel and adjacent, the measured distance between them must lie within a range of ±25% of the reference polygon's width. This tolerance accounts for potential dimensional inconsistencies of polygons and prevents the false detection of segments from other, nearby, or overlapping strips. The final segment $s_{i,1}$, which is always overlapped, is then estimated based on the dimensional properties derived from the reference polygon P_1.

On identifying all segments of P_i, their intersection points are computed to define the polygon's vertices, which are stored as an ordered list. This process iterates until all polygons of the current strip are successfully extracted.

Results and Discussion

Implementation Settings and Results for Background Removal

The data set comprises 100 pairs of input and output images (raster images), where the outputs are manually annotated to ensure high accuracy. The 100 image pairs were selected from 100 distinct cities to ensure maximum geographic coverage across the country, variety in the base cartography style, and drawn polygon dimension and strip development within the API. The selection criteria of the pairs were designed to ensure robust postprocessing during the testing phase by excluding cases that could cause failure, such as the absence of the initial polygon in a strip or the overlapping between different strips. The data set is partitioned into training (60 pairs), validation (15 pairs), and testing sets (25 pairs). To enhance model generalization, the training data set is generated by randomly extracting 512×512-pixel tiles from the selected 60 training pairs, resulting in a total of 8000 tiles. One-half of these tiles are generated using augmentation techniques, including rotation, flipping, and brightness adjustment.

Initially, three distinct models are trained: U-Net with a VGG16 backbone, DeepLabV3 with a ResNet50 backbone, and DeepLabV3 with a ResNet152 backbone (He *et al.* 2016; Ronneberger *et al.* 2015; Simonyan and Zisserman 2015). All models are trained for a maximum of 100 epochs with a batch size of eight and an initial learning rate of 5×10^{-4}. Optimization is performed using the ADAM optimizer with the binary cross-entropy (BCE) loss function (Mannor *et al.* 2005; Kingma and Ba 2015). All experiments are conducted on an NVIDIA Tesla V100S PCIe 32GB GPU.

To assess model performance, the following metrics are used (Equations 5–7):

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} \tag{5}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{6}$$

$$\text{IoU} = \frac{\text{TP}}{\text{TP+FP+FN}} \tag{7}$$

where TP are the correctly segmented polygon pixels, FP are the wrongly segmented polygon pixels, and FN are the wrongly segmented background pixels. Table 1 presents the quantitative results of these initial tests and shows that the DeepLab model with ResNet-152 backbone achieves marginally better performance compared with the other architectures. Therefore, this model is chosen for subsequent experiments.

Moreover, to address the class imbalance issue, where polygon pixels are significantly outnumbered by background pixels, the dice loss function is introduced (Sudre *et al.* 2017). In this study's experiments, three distinct training strategies are evaluated: using only the BCE loss; using only the dice loss; a combined approach integrating both BCE and dice losses. Results demonstrate that using the dice loss function effectively mitigates class imbalance issues, consequently improving the background removal results (Table 2).

Table 1. Metrics from the background removal step using different architectures. The best metrics obtained are highlighted in bold

	Sensitivity (%)	Precision (%)	IoU (%)
U-Net_VGG16	79.28	79.31	65.25
DeepLab_ResNet50	**79.30**	79.08	65.46
DeepLab_ResNet152	78.13	**80.52**	**65.96**

IoU = intersection of union.

Table 2. Metrics from the background removal step using different loss functions. The best metrics obtained are highlighted in bold.

	Sensitivity (%)	Precision (%)	IoU (%)
DeepLab_BCE	78.13	80.52	65.96
DeepLab_DICE	78.67	**81.54**	66.98
DeepLab_BCE_DICE	**81.52**	80.21	**67.88**

IoU = intersection of union.

The proposed RefDeepLab model is then compared with the best-performing model from the previous experiment, DeepLab_BCE_DICE. Both models were trained under identical conditions (i.e., the same data set and hyperparameters). Table 3 and Figure 10 show that RefDeepLab achieves superior performance, demonstrating the effectiveness of the proposed refining blocks integrated into the base DeepLab architecture.

Although the quantitative accuracy metrics are not sufficiently high, a visual inspection of the results indicates that segmentation errors have a minimal effect on the subsequent polygonization process (Figure 9). This robustness can be attributed to two key factors: spatial irrelevance, in which erroneously detected lines are predominantly located along image borders, a region distal to the areas of interest (i.e., the strips); and algorithmic compensation, in which the consistent over- and underestimation of line width is corrected by the application of a thinning algorithm (skeletonization) immediately after segmentation, which reduces all lines to a single-pixel width.

Settings and Results of Arrow Detection

To perform arrow detection, a data set of 370 images (APIs) is created with annotated bounding boxes around each arrow. It is divided into 100 samples for training, 20 for validation, and 250 for testing. Because of the high resolution of the images (widths and heights are mostly between 3500 and 4000 pixels), down-sampling was applied to ensure compatibility with GPU memory constraints during training. The final image size is standardized to 1408×1408 pixels, selected to maintain divisibility by 32 for network architecture requirements.

Quantitative evaluation results demonstrate satisfactory accuracy levels (Table 4, Figure 11). However, a notable number of false arrow detections can be observed, predominantly located near image borders (Figure 12). These erroneous detections are found to have a negligible effect on the overall processing pipeline because of their peripheral positioning, with their exclusion from the subsequent processing steps.

In the few cases when an arrow is not detected, the polygonization process for the corresponding strip cannot be initialized automatically. However, this condition occurs mainly in cases where the APIs present anomalies in the position of the arrows, primarily due to manual transcription errors. These cases are exceptional and constitute a negligible fraction of the data set (ca. 10%). In such a condition, the position of the missing arrow is manually annotated in the YOLO output file to enable subsequent processing.

5.3 Polygonization Results

The evaluation of the polygonization method is conducted by assessing the spatial alignment between detected polygons and their corresponding ground truth instances. A detection is considered a true positive (TP) if the IoU with a ground truth polygon exceeds a predefined threshold; otherwise, it is classified as a false positive (FP). Based on this matching, this study uses two quantitative metrics:

- Precision: ratio of correctly detected polygons (TP) to the total number of detected polygons (TP + FP), quantifying the fidelity

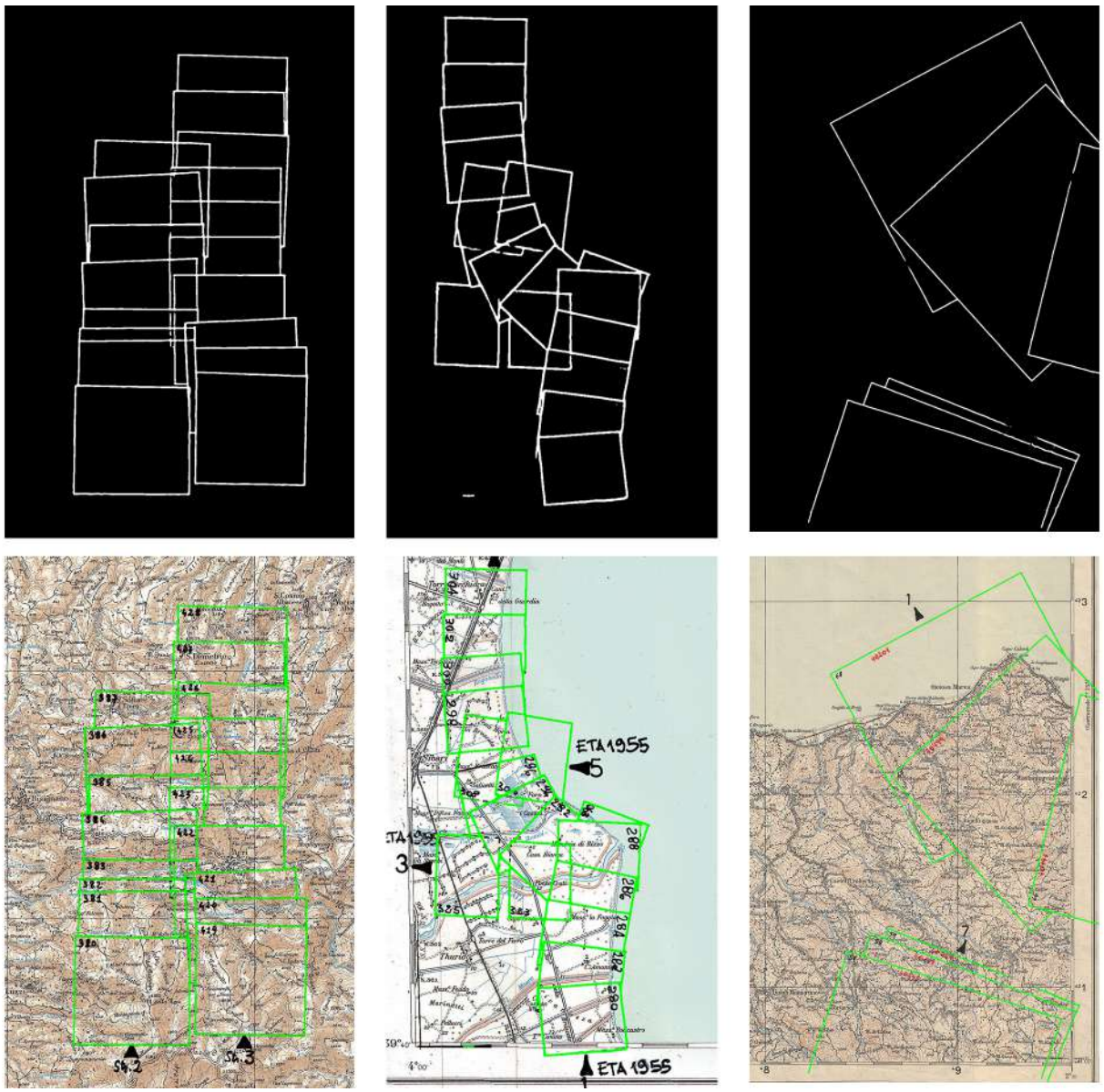

Figure 9. (Top) Background removal results. (Bottom) Results (green lines) superimposed on the original aerial photo index.

of the detection set, measuring the proportion of correct detections among all detections.

- Overall accuracy (OA): ratio of correctly detected polygons (TP) to the total number of ground truth polygons, measuring the completeness of detection, which represents the proportion of ground truth objects that were successfully recovered.

The proposed method was evaluated on a test set of 25 APIs containing 502 ground truth polygons (manually annotated); the approach successfully detected 498 polygons. Quantitative results are computed using IoU thresholds ranging from 0.5 to 0.9 (Table 5). Both precision and OA remain high (>95%) for IoU thresholds up to 0.8, indicating that most polygons are detected with only minor geometric deviations.

However, at an IoU threshold of 0.9, both metrics decreased to approximately 79%. This performance drop is attributed to several factors, including an accumulation of errors from the background removal process. The deformations introduced during skeletonization can adversely affect the Hough transformation's ability to accurately identify line parameters. This is particularly problematic when the initial segmentation produces thick lines, as the skeleton may not reside at the true medial axis. In addition, the fundamental assumption that all polygons within a strip share identical dimensions is sometimes not valid. This inconsistency can lead to differences between the extracted polygons and the originally drawn ones.

Table 5. Quantitative evaluation of the polygonization process.

IoU threshold	Precision (%)	OA (%)
0.5	100	99.20
0.6	99.80	99.00
0.7	99.20	98.41
0.8	96.39	95.62
0.9	79.52	78.88

IoU = intersection of union; OA = overall accuracy.

Figure 13 illustrates selected examples of the complete workflow, and Figure 14 presents some polygonization and vectorization results in different locations across Italy. Furthermore, all the vectorized footprints (output georeferenced shapefiles) are enriched with flight metadata and other attributes manually written on the tables visible on the corresponding API.

The polygon extraction process was applied to the full data set of 4000 APIs, revealing several factors that influence output quality. Optimal performance is achieved when strips do not overlap and polygons are clearly visible, leading to accurate localization. However, parameter adjustments are often necessary to deal with inter-image

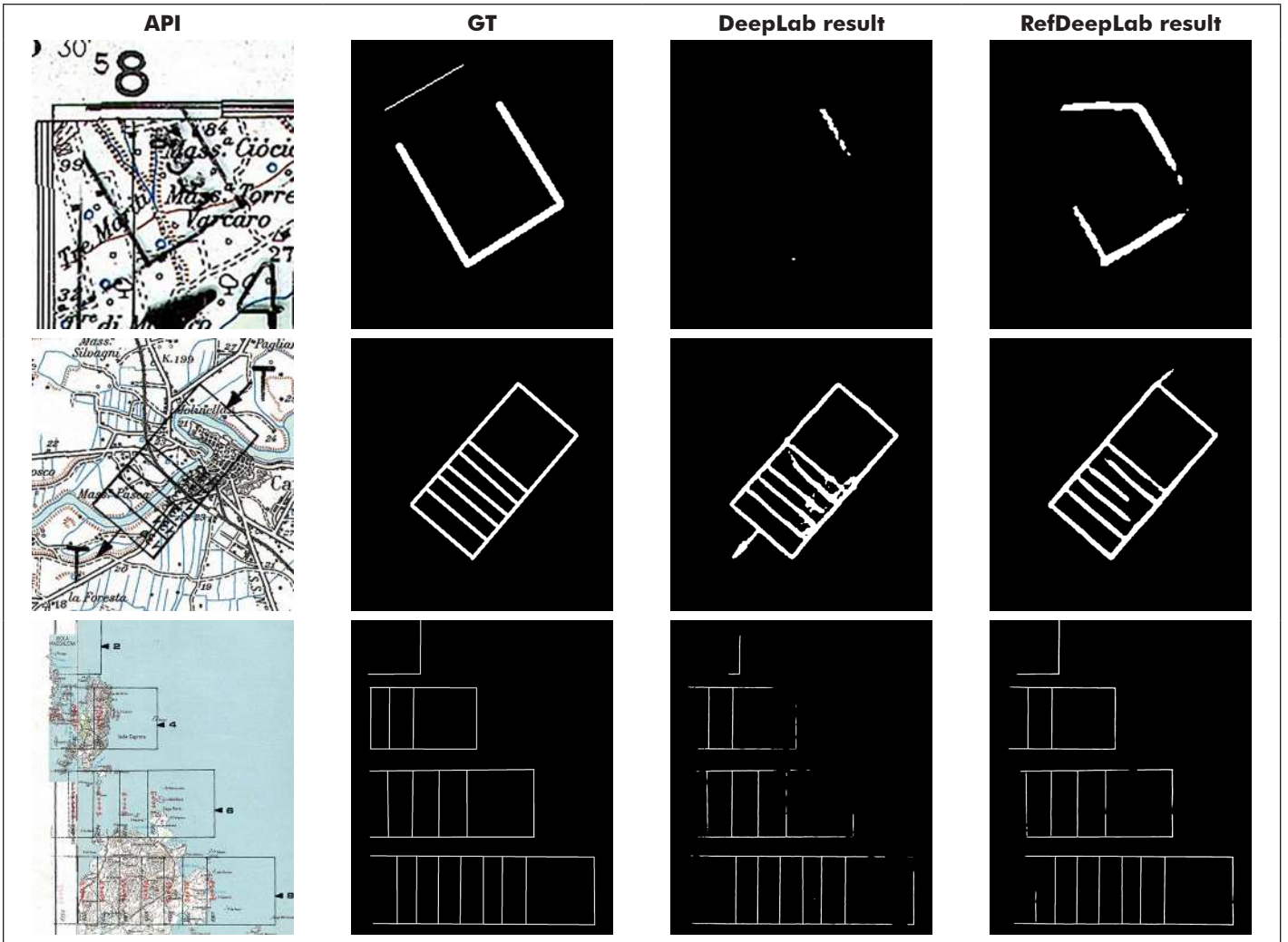

Figure 10. Improvements provided by the proposed RefDeepLab compared with the DeepLabV3. API = aerial photo index.

variations in resolution, polygon size, width-to-height ratios, and the degree of the strips' overlap. The primary challenges affecting quality are categorized as follows.

- Missing or incorrect strip arrows: In some cases, the arrows drawn to indicate the strip direction are either absent or wrongly positioned within the images; to mitigate these issues, their position should be manually fixed.
- No first complete rectangle for the strip: In cases where strips originate at image boundaries, the initial complete polygon is often truncated; to address this limitation, the missing part of the rectangle was manually completed, allowing the polygonization process to proceed.
- Overlapping parallel strips (Figure 15): When two or more parallel strips overlap, the polygonization algorithm encounters topological ambiguity in distinguishing adjacent polygons; in some cases, it is possible to make manual adjustments to guide the algorithm in extracting the polygons.

Conclusions

The paper presents a new methodology for the vectorization of APIs stored in photographic archives. The proposed methodology, applied to thousands of APIs created over the Italian territory, primarily comprises two main stages: automated polygon detection within the scanned APIs and conversion of polygons into vector representations. The work successfully demonstrates how the convergence of computer vision, DL, and computational geometry can address complex heritage vectorization challenges that cannot be solved by any single approach alone.

Table 3. Performance comparison of RefDeepLab and DeepLab. The best metrics obtained are highlighted in bold

	Sensitivity (%)	Precision (%)	IoU (%)
DeepLabV3	81.52	80.21	0.678
RefDeepLab	**81.80**	**81.96**	**0.693**

IoU = intersection of union.

This interdisciplinary methodology represents a significant advancement in automated processing of historical cartographic materials.

Results demonstrate the feasibility and capabilities of the approach to speed up vectorization processes of the image footprints and support online accessibility to historical material. By converting manually compiled finding aids into searchable digital formats, this work removes significant barriers that have limited access to these valuable resources, potentially enabling new forms of historical and environmental research. The presented methodology also addresses the critical preservation of fragile historical material. By creating searchable vector products, the approach ensures long-term preservation of the original data, increases availability, and improves searchability of information reducing at once the required competencies for common users.

Although manual intervention imposed some constraints on the full automation of the process, a significant reduction in the overall vectorization time is achieved. Manually repairing small issues such as missing arrows or initial polygon selection requires only a few seconds per image, whereas manually delineating complete polygons needs up to 1 hr per API. Of the total data set (4000 scanned APIs), approximately 25% (ca. 1000 images) required very minor manual intervention (e.g.,

arrow identification), whereas ca. 150 images necessitated nearly complete manual annotation.

Although the proposed methodology guarantees successful results for Italian National Photographic Archive collections, critical questions remain about its transferability to other national archives with different cartographic conventions, map styles, or preservation conditions. Further validation across diverse archival collections is essential, although API data sets from other countries are not freely available. Moreover, further fine tuning with additional training data is necessary, particularly to improve background removal accuracy and automation of polygons generation. Furthermore, to enhance the robustness of the polygonization process and extract more consistent and coherent line segments, different line-detection techniques such as EDLines or LSD could be adapted as alternatives to the Hough transformation (Cuneyt and Cihan 2011; von Gioi *et al.* 2012). Finally, advances in character recognition and text extraction algorithms could enable the automatic linking of strip metadata to polygons; because of the variability of handwriting in APIs across the decades under study, previously these could be interpreted only manually.

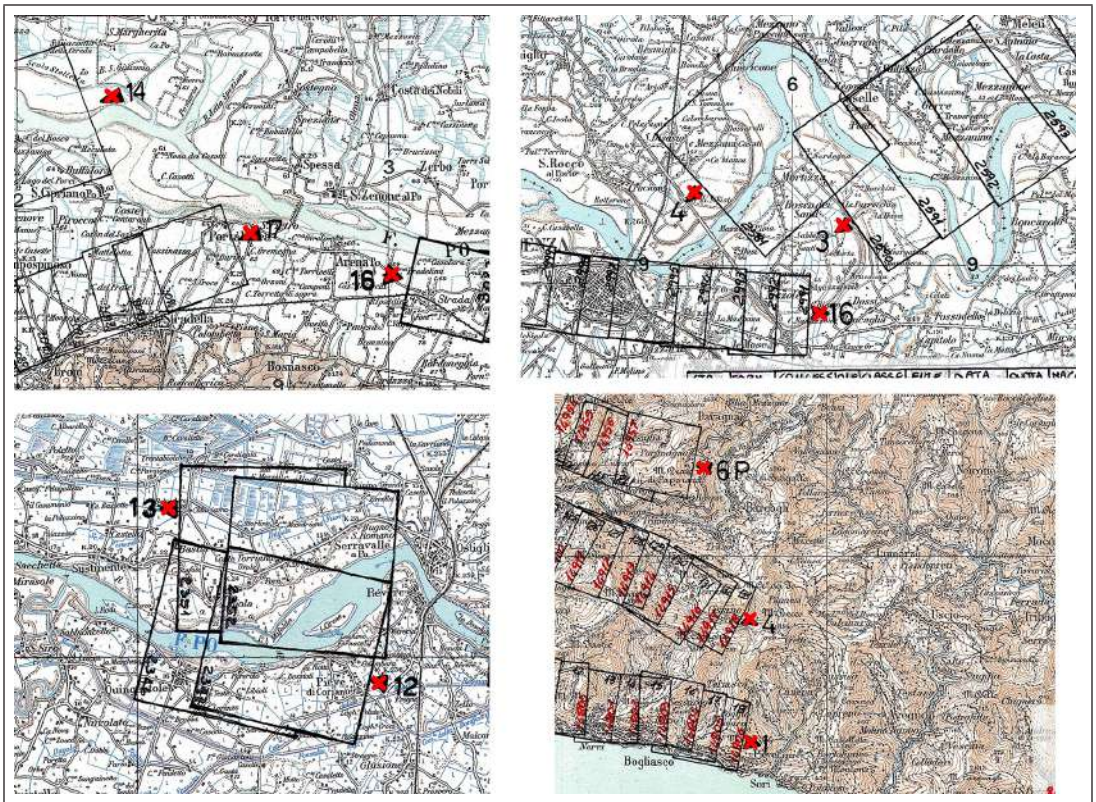

Figure 11. Arrow detection.

Table 4. Metrics from the arrow-detection step.

Total number of arrows	**612**
Correctly detected arrows	574
False detection	109
Sensitivity	84.04%
Precision	93.79%
IoU	79.61%

IoU = intersection of union.

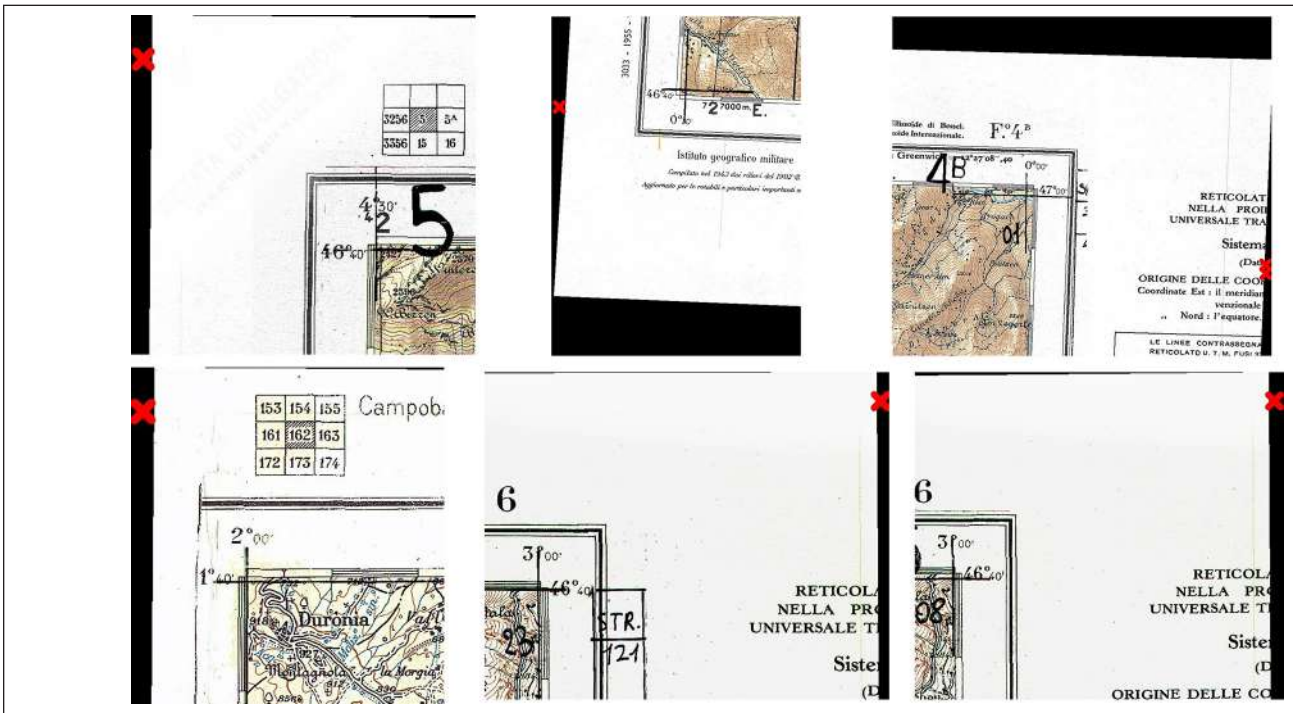

Figure 12. False arrow detection at the borders of images.

References

Akdoğan, C., T. Özer and Y. Oğuz. 2025. PP-YOLO: Deep learning based detection model to detect apple and cherry trees in orchard based on Histogram and Wavelet preprocessing techniques. *Computers and Electronics in Agriculture* 232:110052.

Aman, Ibrahim and J. Saini. 2023. YOLO-NAS-powered forest monitoring system: A New frontier in forest management, *Proceedings of the Seventh International Conference on Image Information Processing (ICIIP)*, 22–24 November 2023, Solan, India(IEEE Computer Society, NJ, USA), pp. 312–315. doi: 10.1109/ICIIP61524.2023.10537669.

Archivio Aerofototeca Nazionale. 2026. National Aerial Photo Archive. https://afn.cultura.gov.it/webgis/ (last date accessed: 5 January 2026).

Armenakis, C., F. Leduc, I. Cyr, F. Savopol and F. Cavayas. 2003. A comparative analysis of scanned maps and imagery for mapping applications. *ISPRS Journal of Photogrammetry and Remote Sensing* 57(5–6):304–314.

Arteaga, M. G. 2013. Historical map polygon and feature extractor, *Proceedings of the 1st ACM SIGSPATIAL International Workshop on MapInteraction Association for Computing Machinery,* Orlando Florida 5 November 2013, (Association for Computing Machinery: New York, NY, United States), , pp. 66–71. https://doi.org/10.1145/2534931.2534932

Ballard, D. H. and C. M. Brown. 1982. *Computer Vision.* (Prentice Hall: New Jersey, USA), 544p.

Bay, H., T. Tuytelaars and L. Van Gool. 2006. SURF: Speeded up robust features *Proceedings of the European Conference on Computer Vision 3951*, 7–13 May 2006, Graz/Austria (Springer-Verlag: Berlin: Heidelberg, Germany), pp. 404–417.

Bochkovskiy, A., C. Wang and H. M. Liao. 2020. YOLOv4: Optimal speed and accuracy of object detection. arXiv:2004.10934. https://doi.org/10.48550/arXiv.2004.10934.

Bundesamt für Eich- und Vermessungswesen. 2026. Austrian Map. https://www.bev.gv.at (last date accessed: 5 January 2026).

Cantoro, G. 2014. Speeding up georeferencing with subpixel accuracy, *Proceedings of the CAA 2012—Archaeology in the Digital Era,* 26–30 March 2012, *Southampton*UK. (Amsterdam University Press: Netherland), pp. 752–760.

Cantoro, G. 2017. Handling hundreds of aerial images without ground references: Archaeological photointerpretation in the era of digital photogrammetry. *Archaeologia Aerea* XI:23–29.

Chen, L., G. Papandreou, L. Kokkinos, K. Murphy and A. L. Yuille. 2018. Semantic image segmentation with deep convolutional nets, Atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40:834–848.

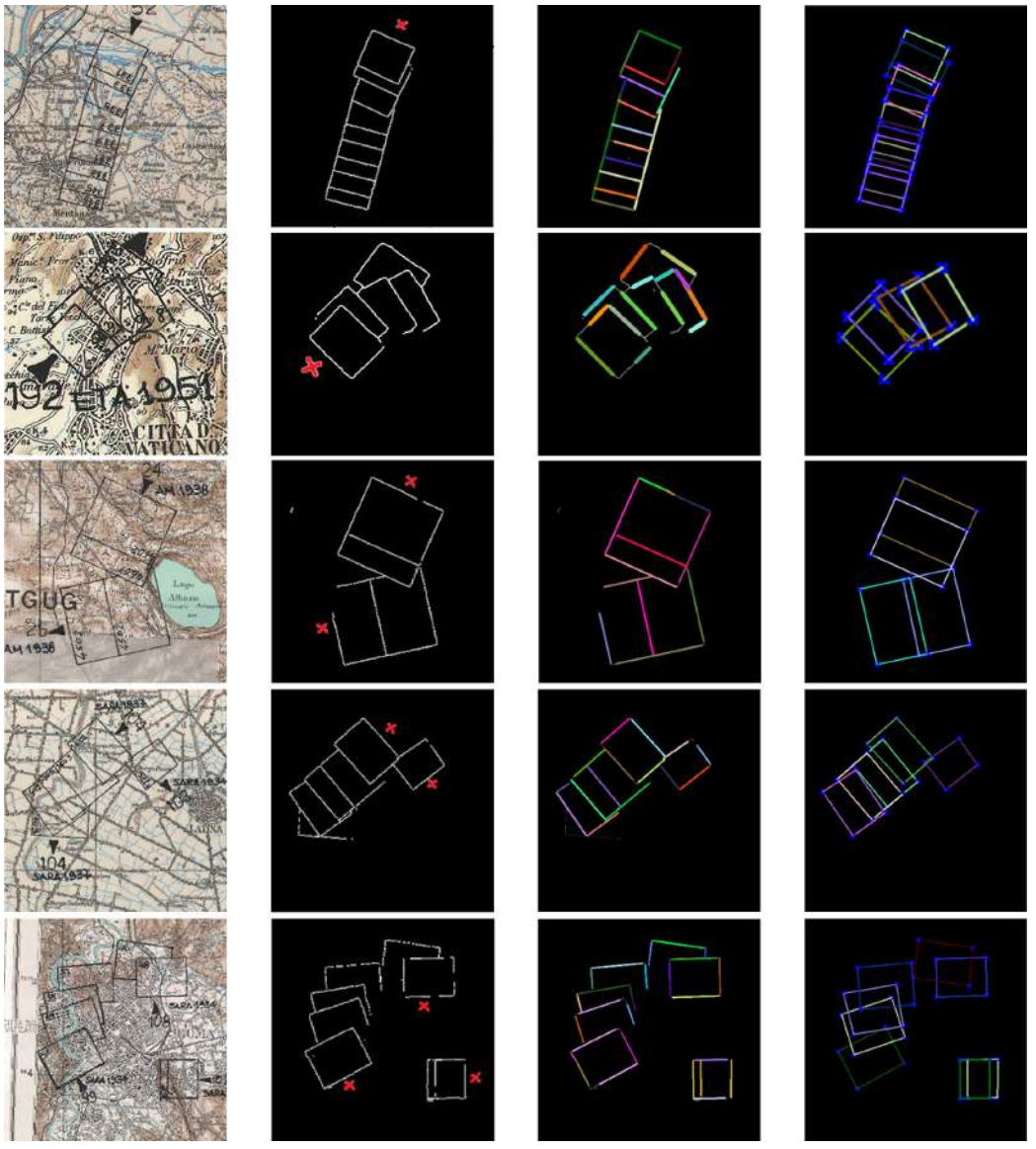

Figure 13. Obtained results: (a) crop from origin image, (b) background removal + skeletonization + arrow detection, (c) lines detection, (d) final result of the polygonization.

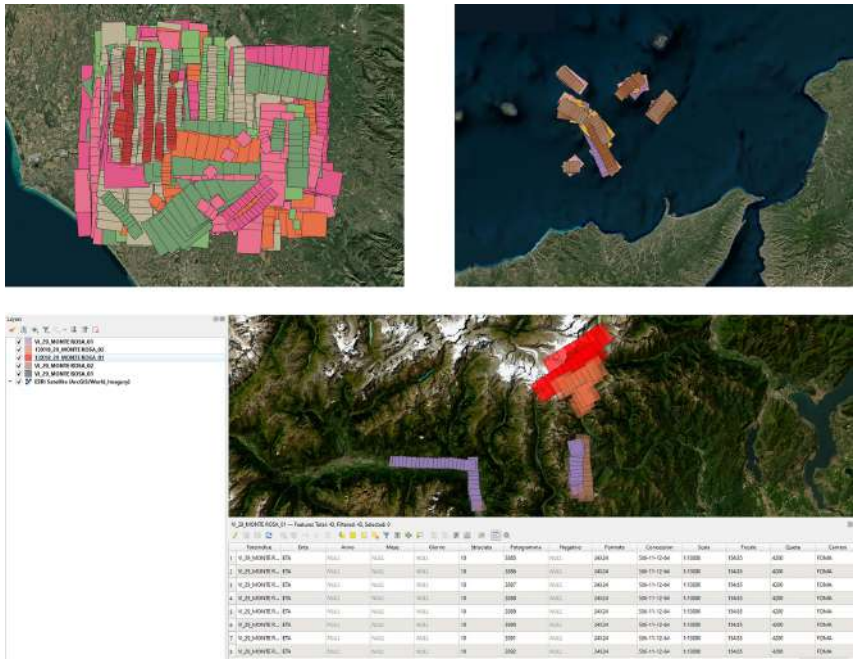

Figure 14. Vectorization results enriched with metadata for each aerial photo index.

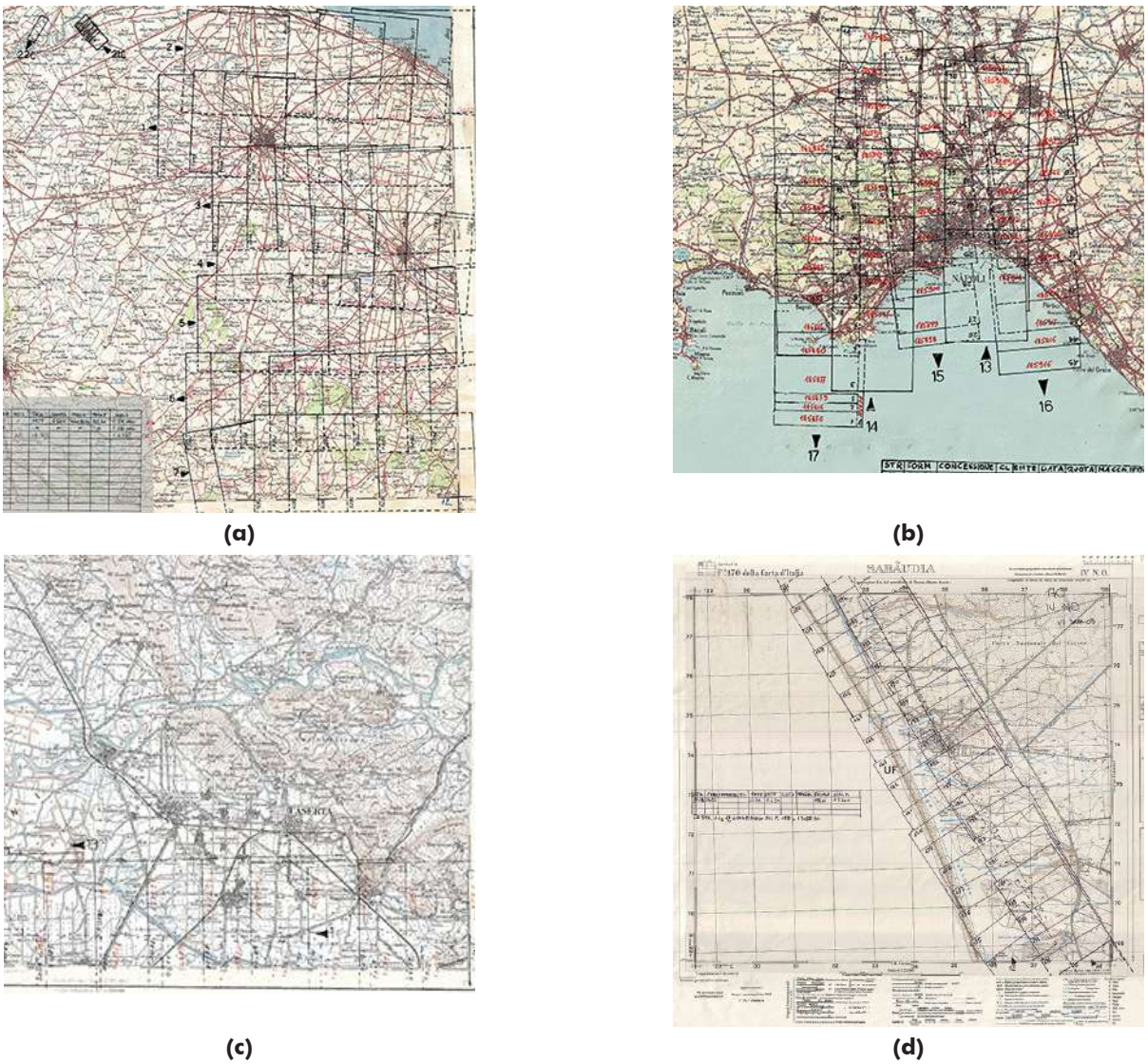

Figure 15. Complex aerial photo indexes with (a, b) parallel overlapping strips, (c) of anomalous arrows position within the strips, and (d) of truncated first polygons.

Chen, Y., E. Carlinet, J. Chazalon, C. Mallet, B. Duménieu and J. Perret. 2021. Vectorization of historical maps using deep edge filtering and closed shape extraction, *Proceedings of the International Conference on Document Analysis and Recognition*, X–X Month XXXX, Lausanne, Suitzerland (Springer: Germany), pp. 510–525.

Chen, Y., J. Chazalon, E. Carlinet, M. Ôn Vũ Ngoc, C. Mallet and J. Perret. 2024. Automatic vectorization of historical maps: A benchmark. *Plos One* 19(2):e0298217.

Chen, Z., B. Chen, Y. Huang and Z. Zhou. 2025. GE-YOLO for weed detection in rice paddy fields. *Applied Sciences* 15(5):2823.

Chiang, Y. Y. and C. A. Knoblock. 2006. Classification of line and character pixels on raster maps using discrete cosine transformation coefficients and support vector machine, *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06) 2*, 20–24 August 2006, Hong Kong, China (IEEE Computer Society: USA), pp. 1034–1037.

Chiang, Y. Y. and C. A. Knoblock. 2008. Automatic extraction of road intersection position, connectivity, and orientations from raster maps, *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 5–7 November 2008, City, State/Country (Association for Computing Machinery: New York, NY, United States), pp. 1–10.

Chiang, Y. Y. and C. A. Knoblock. 2009. An approach to automatic road vectorization of raster maps, *Proceedings of the 8th IAPR International Workshop on Graphics RECognition (GREC'09)*, 22–23 July 2009, La Rochelle, France (Springer: Germany), pp. 1–3.

Chiang, Y. Y., S. Leyk and C. A. Knoblock. 2013. Efficient and robust graphics recognition from historical maps, *Proceedings: Graphics Recognition. New Trends and Challenges (GREC 2011) Seoul, Korea 7423*, 15–16 Sptember 2011, Seoul, Korea (Springer: Germany), pp. 25–35.

Chrysovalantis, D. G. and T. Nikolaos. 2020. Building footprint extraction from historic maps utilizing automatic vectorisation methods in open source GIS software, *Proceedings of the International Workshop on Automatic Vectorisation of Historical Map,* 13 March 2020, Budapest (iInternational Cartographic Association), pp. 9–17.

Craciun, D. and A. Le Bris. 2022. Automatic algorithm for georeferencing historical-to-nowadays aerial images acquired in natural environments. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 43:21–28.

Cuneyt, A. and T. Cihan. 2011. EDLines: A real-time line segment detector with a false detection control. *Pattern Recognition Letters* 32:1633–1642.

Czechoslovakia Government, Archives. 2026. Land Surveying Office. https://lms.cuzk.cz/ (last date accessed: 5 January 2026).

DeKruger, D. and B. R. Hunt. 1994. Image processing and neural networks for recognition of cartographic area features. *Pattern Recognition* 27(4):461–483.

Deng, H., S. Zhang, X. Wang., T. Han and Y. Ye. 2025. USD-YOLO: An enhanced YOLO algorithm for small object detection in unmanned systems perception. *Applied Sciences* 15(7):3795.

Deng, J., W. Dong, R. Socher, Li, L., K. Li and L. Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 20-25 June 2009, Miami, Florida, USA (IEEE Computer Society: USA), pp. 248–255.

Department of Cadastre and Surveying. 2026. e-Cadastre. https://eservices.dls.moi.gov.cy/ (last date accessed: 5 January 2026).

Douglas, D. H. and T. K. Peucker. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Canadian Cartographer* 10:112–122.

Dunesme, S., H. Piégay and S. Mustière. 2022. Automatic vectorization of fluvial corridor features on historical maps to assess riverscape changes. *Cartography and Geographic Information Science* 49(6):512–527.

Farella, E. M., L. Morelli, F. Remondino, J. P. Mills, N. Haala and J. Crompvoets. 2022. The EuroSDR TIME benchmark for historical aerial images. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 43:1175–1182.

Federal Office of Topography, Swisstopo. 2026. Swisstopo—Knowing Where To Go. https://www.swisstopo.admin.ch (last date accessed: 5 January 2026).

FOTOTECA. https://fototeca.cnig.es/fototeca/. (last date accessed: 29 January 2026).

Gede, M., V. Árvai, G. Vassányi, Z. Supka, E. Szabó, A. Bordács, C. G. Varga and K. Irás. 2020. Automatic vectorisation of old maps using QGIS–tools, possibilities and challenges, *Proceedings of the International Workshop on Automatic Vectorisation of Historical Map,* 13 March 2020, Budapest (iInternational Cartographic Association), We added the missing information, pp. 37–44.

Gonzalez, R. C. and R. E. Woods. 2008. *Digital Image Processing,* 3rd ed. Pearson Prentice Hall: new Jersey/USA.

Hancer, E. and R. Samet. 2011. Advanced contour reconnection in scanned topographic maps, *Proceedings of the 5th International Conference on Application of Information and Communication Technologies (AICT),* 12-14 October 2011, Baku, Azerbaijan (IEEE Computer Society: USA), pp. 1–5.

Hatcher, W. G. and W. Yu. 2018. A survey of deep learning: Platforms, applications and emerging research trends. *IEEE Access* 6:24411–24432.

He, K., X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 27 30 June 2016, Las Vegas, Nevada, USA (IEEE Computer Society, USA), pp. 770–778.

Henderson, T. C. and T. Linton. 2009. Raster map image analysis, *Proceedings of the 10th International Conference on Document Analysis and Recognition,* 26-29 July 2009, Barcelona, Spain (IEEE Computer Society, USA), pp. 376–380.

Hori, O. and A. Okazaki. 1992. High quality vectorization based on a generic object model. In *Structured Document Image Analysis*, edited by H.S. Baird, H. Bunke and K. Yamamoto. Berlin, Heidelberg: Springer, pp. 325–339. https://doi.org/10.1007/978-3-642-77281-8_15 (last date accessed: 5 January 2026).

Hough, P. V. 1959. Machine analysis of bubble chamber pictures, *Proceedings of the International Conference on High Energy Accelerators and Instrumentation*, 14-19 September 1959, Cern, Geneva/Switzerland (Current Science Association: India), pp. 554–556.

Ikechukwu, S. and E. Akin. 2022. High performance network for detection of surface defects on hot-rolled steel strips based on an optimized YOLO V3, *Proceedings of the 9th International Conference on Electrical and Electronics Engineering (ICEEE),* 29-31 March 2022, Alanya, Turkey (IEEE Computer Society: USA), pp. 1–6.

Iosifescu, I., A. Tsorlini and L. Hurni. 2016. Towards a comprehensive methodology for automatic vectorization of raster historical maps. *e-Perimetron* 11(2):57–76.

Itonaga, W., I. Matsuda, N. Yoneyama and S. Ito. 2003. Automatic extraction of road networks from map images. *Electronics and Communications in Japan (Part II: Electronics)* 86(4):62–72.

Jain, R., R., Kasturi, R. and B. G. Schunck. 1995. *Machine Vision*, 5th ed. McGraw-Hill: New York, pp. 309–364.

Janssen, R. D. and A. M. Vossepoel. 1997. Adaptive vectorization of line drawing images. *Computer Vision and Image Understanding* 65(1):38–56.

Jiao, C., M. Heitzler and L. Hurni. 2022. A fast and effective deep learning approach for road extraction from historical maps by automatically generating training data with symbol reconstruction. *International Journal of Applied Earth Observation and Geoinformation* 113:102980.

Jiao, C., M. Heitzler and L. Hurni. 2024. A novel framework for road vectorization and classification from historical maps based on deep learning and symbol painting. *Computers, Environment and Urban Systems* 108:102060.

Jin, Z., J. Duan, L. Qiao, T. He, X. Shi and B. Yan. 2025. MTGS-Yolo: A task-balanced algorithm for object detection in remote sensing images based on improved yolo. *Journal of Supercomputing* 81:542.

Kang, M., F. F. Ting, R. C. Phan and C. Ting. 2025. PK-YOLO: Pretrained knowledge guided YOLO for brain tumor detection in multiplanar MRI slices, *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 26 Feb.-06 March 2025 to March 6 2025,, Tucson, Arizona, USA (IEEE Computer Society, USA), pp. 3732–3741.

Karabork, H., B. Kocer, I. O. Bildirici, F. Yildiz and E. Aktas. 2008. A neural network algorithm for vectorization of 2D maps. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XXXVII Part B2:473–480.

Kingma, D. and J. Ba. 2015. Adam: A method for stochastic optimization, *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 07-09 May 2015, San Diego, California, USA (Publisher: Publisher location), pp. 01–15.