ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF COMPUTER SCIENCE
AND ENGINEERING

ARTIFICIAL INTELLIGENCE

MASTER THESIS

in

Machine Learning for Computer Vision

AUTOMATIC DETECTION OF THE INFERIOR ALVEOLAR NERVE IN CT SCANS WITH CENTERNET

CANDIDATE	SUPERVISOR
Alessandro Dicosola	Prof. Samuele Salti
	CO-SUPERVISOR
	Prof. Gerardo Pellegrino

Academic year 2020-2021

Session 3rd

Contents

Abstract

The *inferior alveolar nerve* (IAN) lies within the *mandibular canal*, named *inferior alveolar canal* in literature. The detection of this nerve is important during maxillofacial surgeries or for creating dental implants. The poor quality of cone-beam computed tomography (CBCT) and computed tomography (CT) scans and/or bone gaps within the mandible increase the difficulty of this task, posing a challenge to human experts who are going to manually detect it and resulting in a time-consuming task. Therefore this thesis investigates two methods to automatically detect the IAN: a non-data driven technique and a deep-learning method. The latter tracks the IAN position at each frame leveraging detections obtained with the deep neural network CenterNet, fined-tuned for our task, and temporal and spatial information.

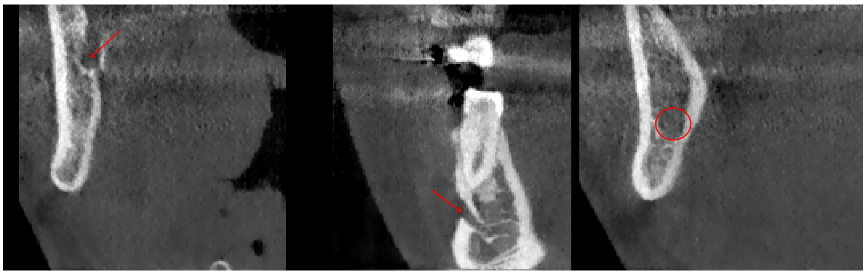
Chapter 1

Introduction

In the mandibular canal, named inferior alveolar canal by researchers [13], resides the inferior alveolar nerve, hereinafter IAN, along with arteries and veins, starting from the Spix spine (Figure 1.1a) and ending in the mental foramen (Figure 1.1b). The identification of this nerve is important in order to avoid damages during maxilofacial surgeries [29] or plants placement [3]. Therefore, before surgery, its position must be estimated with accurate precision. This task is usually done manually by radiographers therefore it's time-consuming and prone to errors. Moreover, bone voids within the mandible, whose density is similar to the mandibular canal one, hide the real position of the IAN (Figure 1.1c), requiring to follow the canal, frame by frame, slowing down the task completion and increasing the chances of errors.

AI is widely used in the medical domain: analysis of images through segmentation ([31]) in order to localize lesions or malformations, diseases diagnosis aided through classifications [16] and drug discovery [18] to name a few.

Due to the nature of the media used for executing IAN detection (CT scan),



(a) Spix spine. (b) Mental foramen. (c) IAN in the middle.

Figure 1.1: Coronal view of a CT scan.

computer vision seems ideal for coping with this problem, in particular computer vision techniques based on machine and deep learning. Despite being powerful, machine learning techniques require handcrafted features and heavy preprocessing steps therefore deep learning seems more suitable due to its ability to find patterns using raw features, in this case pixels. Hence convolutional neural networks (CNN) are preferred, which reached very high performances on many specific tasks (image classifications [8], object detections [19], super resolution [24], image and panoptic segmentation [31], depth estimation [32], ...) within different domains.

Despite segmentation being an obviuos choice due to different shapes that IAN could have in its pathway [2], in this thesis we explored an alternative approch that is object detection: we are not interested in finding, precisely, voxel belonging to the nerve but we would like to extract its position frame by frame.

1.1 Related works

Kainmueller et al. [15] computed a surface of the mandibular bone using statistical shape modelling on top of which a graph is constructed sampling points on equidistant planes where each weight edge between sampled points is computed considering image intensity inside a cylinder at sample points within an inner radius ($r < r_i$), within a border ($r_i < r < r_b$) and outside the border ($r > r_b$). Then a Dijkstra-based optimization is applied in order to find the path from source to target in order to detect the ”darkest tunnel” surrounded by interesting bright borders. For this task mandible bone voxel annotation are required which means more data to annotate.

Jaskari et al. [14] and Bayrakdar et al. [5] trained a 3D U-Net-like convolutional network [28] for segmenting voxel inside the input volume. The former was able to surpass Kainmueller et al. [15] obtaining distances in millimeters around $0.5\,\mathrm{mm}$ for 90% of the primary test data as well as robustness to label noise.

Liu et al. [21] using segmentation over the axial plane is able to extract a region of interest that encloses the mandibular canal (MC) and the mandibular third molar (M3) which is used for cropping CBCT scans. The cropped region is, then, segmented by a U-Net convolutional network.

Kwak et al. [17] studied IAN segmentation with different 2D and 3D convolutional neural network (SegNet[4], U-Net, 3D U-Net [6]) showing how difficult this task can be due to noise and different shapes.

Chapter 2

Methods

Different from works presented in Section 1.1, we tackled on this problem as an object detection one, following clinical indications, since the precise segmentation is not required to plan most routinely performed interventions. Hence, localization of the center of the canal is sufficient in most practical cases.

In the following sections we are going to present the two methods used for executing this task: the former is based on a non-data driven technique which model the canal as a deformable linear object (DLO) and the latter exploits detections found by a model for object detection, CenterNet, fine-tuned for this specific problem.

2.1 Background

2.1.1 Ariadne

Ariadne [11] identifies the parameters of Deformable Linear Objects (DLOs) with a predefined model (b-spline) trying to solve cluttering and occlusion (as well as self-occlusion) over simple and complex backgrounds.

The endpoints detection for each DLOs is meant to be an external step. Ariadne uses a convolutional neural network (YoloV2 [25]) finetuned on the Eletrical Cable Dataset [1].

The core algorithm find the best path (*walk*) over an adjaceny graph built using superpixels.

The superpixels are created using SLIC [1] which apply a modified k-means clustering on vectors defined by colors in the CIELAB color space and spatial position (x,y). Each point is compared with the nearest centroid whose search window overlap its position. Moreover, the distance metric used takes into account the color similarity and the spatial proximity is weighted by a parameter that controls the compactness of each superpixel allowing either compact clusters and/or visually uniform clusters. A graph $G = (V, E)$ is built, where each region, defined by the vertex v_i (centroid), is connected with adjacent ones, through an edge $e_{i,k}$ (distance vector between two vertices).

The *walk* (π_i) starts from one seed point (one of the two endpoints found previously) and iteratevely choses the point v_n (Figure 2.1) that maximize

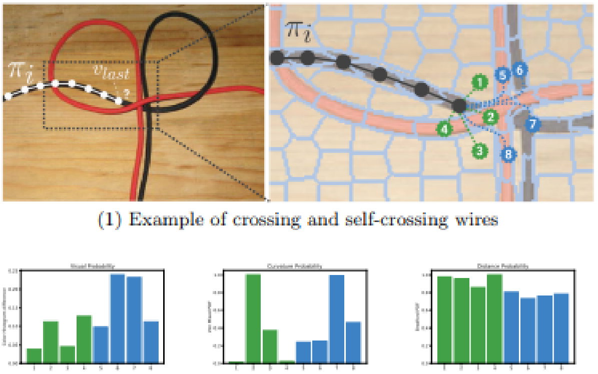
$$p(\hat{\pi}_{i,v_n}|\pi_i) = p_V(\hat{\pi}_{i,v_n}|\pi_i) \cdot p_C(\hat{\pi}_{i,v_n}|\pi_i) \cdot p_D(\hat{\pi}_{i,v_n}|\pi_i) \tag{2.1}$$

where $\hat{\pi}_{i,v_n}$ is the path π_i with vertex v_n and $p_V(\cdot)$, $p_D(\cdot)$, $p_C(\cdot)$ are *Visual*, *Distance*, *Curvature* likelihood. The *Visual* and *Distance* likelihood are computed normalizing a distance using the Bradford normal distribution:

$$p(\hat{\pi}_{i,v_n}|\pi_i) = \frac{c_v}{log(1 + c_v)(1 + c_v(1 - d))} \tag{2.2}$$

where d is respectively the intersection between two color histograms in the HSV color space and the spatial distance. The *Curvature* likelihood

[1] `https://github.com/m4nh/cables_dataset`



(1) Example of crossing and self-crossing wires

(2) Visual likelihood (3) Curvature likelihood (4) Distance likelihood

Figure 2.1: One step of the algorithm used in Ariadne [11, Fig. 3] where given a current *walk* different points are considered for selecting the most probable one.

considers the angle difference between two consecutive edges in order to prioritize vertices which allow a smooth path from start to end. It is computed with the following equation:

$$p_C(\hat{\pi}_{i,v_n}|\pi_i) = \prod_a M(\frac{\phi_a - \phi_{a+1}}{2}, m) \tag{2.3}$$

where ϕ_i is the angle between two consecutive edges and $M(\cdot)$ is the von Mises distribution.

A *walk* is ended if it reaches another seed or the distance from the vertex and the other seed is smaller than a threshold. The best *walk* is the one whose curvature likelihood is maximum.

2.1.2 Object detection

Object detection task aims to find for each object in a given image a set of properties: class and bounding box.

Two stage object detectors at first were using proposals found with Selective Search [10, 9] then with Region Proposal Network [26] Those were forwarded to a network, after being wrapped (in case of patches extracted from the original image) or transformed by a RoI Pooling layer (in case of patches extracted from features computed by the features extractor), to classify them and regress their bounding-box corrections. At the end, detections are filtered with non maxima suppression over bounding boxes intersection-over-union (IoU) to remove overlapping ones.

Object detection is inherently unbalanced: the amount of easy negative samples (background) is greater than the positive ones decreasing the performance of the model to distinguish between negatives and positives. This is solved using top proposals in two stage detectors and hard negative mining (that is, for training are used only negative samples whose classification loss is the highest in order to let the model focus on hard negative examples) in one stage detectors [22].

In order to solve this problem, the FocalLoss [20] was proposed for coping automatically with class unbalance down-weighting the loss when easy negatives and easy positives are classified focusing the training on hard positives and negatives. The FocalLoss has the following form:

$$FL(p_t) = -\alpha_t(1-p_t)^\gamma \log(p_t) \tag{2.4}$$

where

$$p_t = \begin{cases} p & y = 1 \text{ positive} \\ 1-p & \text{otherwise} \end{cases} \tag{2.5}$$

$$a_t = \begin{cases} \alpha & y = 1 \text{ positive} \\ 1-\alpha & \text{otherwise} \end{cases} \tag{2.6}$$

p is the model's estimated probability, γ is a hyperparameter for controlling the down-weighting factor and α is a further hyperparameter for considering the class unbalance.

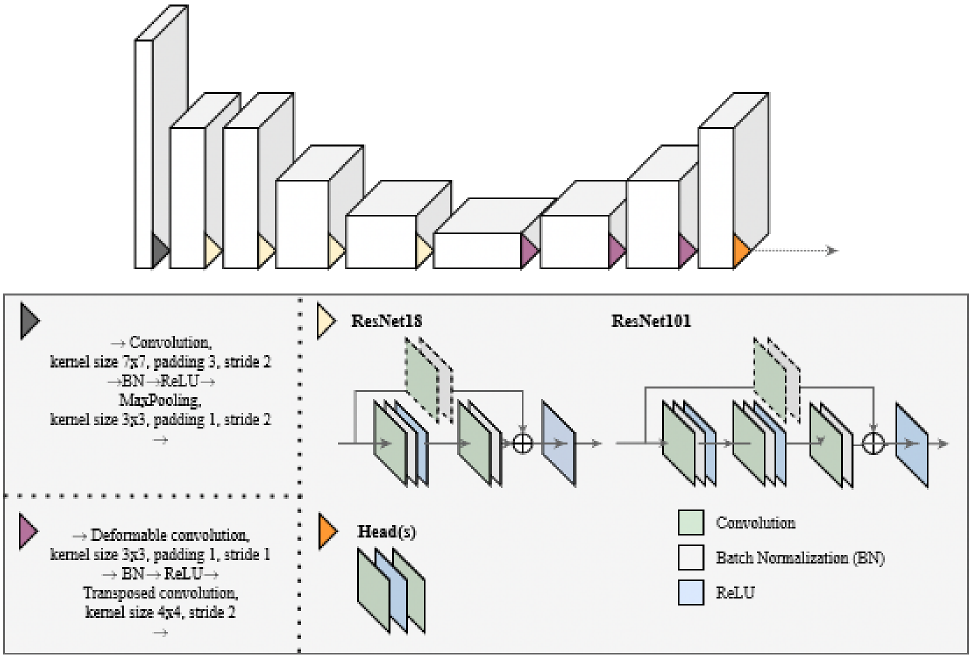

→ Convolution,
kernel size 7x7, padding 3, stride 2
→BN→ReLU→
MaxPooling,
kernel size 3x3, padding 1, stride 2
→
ResNet18
ResNet101
→ Deformable convolution,
kernel size 3x3, padding 1, stride 1
→ BN → ReLU→
Transposed convolution,
kernel size 4x4, stride 2
→
Head(s)
Convolution
Batch Normalization (BN)
ReLU

Figure 2.2: High level graphical overview of CenterNet. Gray parallelepipeds represent activations after an operation depicted with a colored triangle. The width and height of each parallelepiped represents, respectively, the channels and spatial dimensions after each operation. A full description is presented in Section 2.1.3.

2.1.3 CenterNet

CenterNet[33] is an end-to-end differentiable *encoder-decoder* network that allows to carry on the object detection task (and many others) efficiently and more simply than two or one stage object detectors based on proposals [25] because does require neither proposals nor filtering final detections since only one prediction per object is estimated.

Encoder: ResNet In this work we used CenterNet with ResNet18 and ResNet101 as backbone, also called encoder, which has to extract meaningful features, then used by the decoder. ResNet [12] introduce the *residual connection* in order to let the network learn the identity function $G(x) = x + F(x)$ where x is the input (*identity*), $F(\cdot)$ is the *residual function* to learn and $G(\cdot)$ is the output forwarded to a non-linear activation function. All of them define a *residual block*. As a consequence the loss landscape is smoother allowing the training of very deep networks. The input spatial dimensions are reduced by the downsample block (*gray triangle*, see Figure 2.2) by a factor 4, then a sequence of residual blocks (*yellow triangle*, see Figure 2.2) at each level extract features from previous activations halving the spatial dimension and doubling the channels (except for the first level): in particular ResNet18 apply two *residual blocks* at each level; instead ResNet101 applys at the first level 3 *residual blocks*, than 4, next 23 and at the end 3. Moreover, the *residual block* is different for ResNet18 and ResNet101, as shown in Figure 2.2: the former uses only two 3×3 convolutions instead the latter uses a 3×3 convolution preceded and followed by 1×1 convolutions that reduce then expand the amount of channels; this is required due to the exponential increase of them. In both cases, when the output of a block is the input of a residual block that halves the spatial dimension, 3×3 convolution with stride 2 in the skip connection path downscale the input dimensions.

Decoder The features extracted by the backbone are then processed by the decoder which is composed by a sequence of 1×1 *deformable convolutions* [7] followed by 3×3 *transposed convolutions*[30] (Figure 2.2): the former allow learning offsets in order to apply convolutions with sparse kernels, instead of dense ones as in normal convolutions, adapting it based on the object scale, the latter allow upscaling activations with a learnable function instead of a non-learnable one (e.g. bilinear upsampling). The spatial dimensions are double and channels halved each time a transposed convolution is applied.

Heads Based on the task, CenterNet could have many heads. For a simple object detection task there are the *heatmap head* which contains at each pixel position the probability that it is a center, *offset head* which allow to correct the stride position extracted from the heatmap and *size head* which allow to extract the width and height of bounding box. In particular a 3×3 convolution follow a 1×1 one for generating output tensors in the strided resolution (Figure 2.2).

Instead of using anchors, bounding boxes with fixed area and aspect ratio, CenterNet detect objects based on their center point $p = (x, y)$, thus it is neither required to classify proposals extracted by RPN (using anchors) or Selective Search, as in two-stage detector, nor classify each anchor at each position in the final activation, as in one stage detector: this allows a faster inference. Moreover, since each object can be detected only once filtering out overlapping detections is avoided.

In order to train the network each center position p of class c is downscaled by a factor D, then $\tilde{p} = \left\lfloor \frac{p}{D} \right\rfloor$ is computed. Hence, the ground truth heatmap $H \in R^{\frac{H}{D} \times \frac{W}{D} \times C}$ is created generating unnormalized Gaussians at each $\tilde{p}$ with an object-size adaptive σ_p at index c taking the element-wise maximum if two centers overlap on the same class:

$$Y_{xyc} = e^{-\frac{(x-\tilde{p}_x)^2+(y-\tilde{p}_y)^2}{2\sigma_p^2}} \tag{2.7}$$

Therefore given an image $I \in R^{H \times W \times 3}$ CenterNet predicts a heatmap $\hat{H} \in R^{\frac{H}{D} \times \frac{W}{D} \times C}$, an offset map $\hat{O} \in R^{\frac{H}{D} \times \frac{W}{D} \times 2}$ for correcting the positions due to the stride and bounding box sizes $\hat{S} \in R^{\frac{H}{D} \times \frac{W}{D} \times 2}$. Then *non-maxima suppression* (NMS) is applied on the heatmap for extracting peaks in a neighborhood 3×3. At each peak position $\hat{p}_k$ for class c, the probability $\hat{y}_{xyc}$, offset $\hat{o}_{p_k}$ and size $\hat{s}_{p_k}$ are extracted, respectively, from $\hat{H}$, $\hat{O}$, $\hat{S}$ and the total loss is computed (thus the supervision is done only on peak positions):

$$\mathcal{L} = \mathcal{L}_k + \lambda_{size}\mathcal{L}_{size} + \lambda_{offset}\mathcal{L}_{offset} \tag{2.8}$$

where L_{size} is

$$L_{size} = \frac{1}{N}\sum_{k=1}^{N} |\hat{s}_{p_k} - s_{p_k}|$$

and s_{p_k} is the ground truth size, L_{offset} is

$$L_{offset} = \frac{1}{N}\sum_{k=1}^{N} |\hat{o}_{p_k} - \left(\frac{p_k}{D} - \tilde{p_k}\right)|$$

and L_k is the pixel-wise FocalLoss:

$$\mathcal{L}_k = \frac{1}{N}\sum_{x,y,c} \begin{cases} (1-\hat{y}_{xyc})^{\alpha} \cdot -\log \hat{y}_{xyc} & y_{xyc} == 1 \\ (1-y_{xyc})^{\beta} \cdot (\hat{y}_{xyc})^{\alpha} \cdot -\log(1-\hat{y}_{xyc}) & otherwise \end{cases} \tag{2.9}$$

which solves class unbalance and where

1. N is the numbers of keypoint in the ground truth heatmap

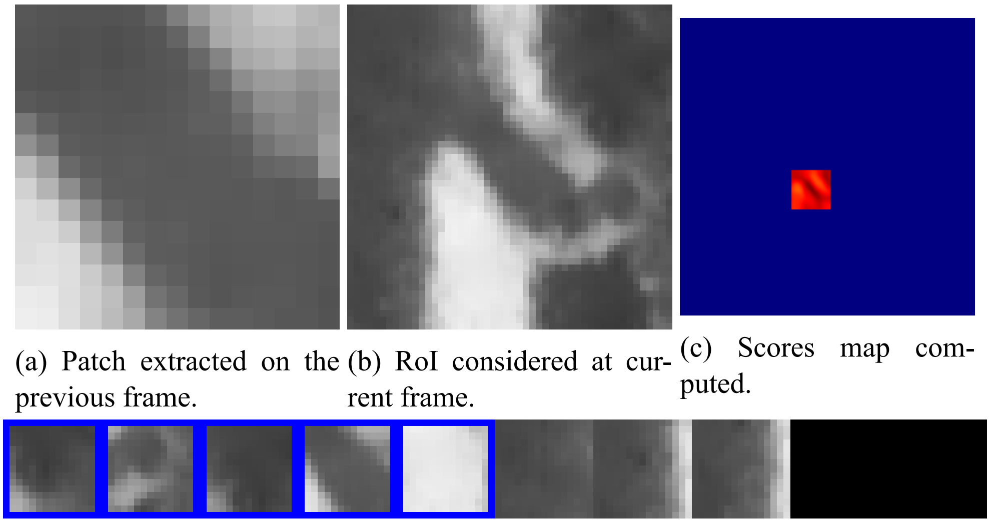
2. $\boldsymbol{(\hat{y}_{xyc})^{\alpha}}$ and $\boldsymbol{(1-\hat{y}_{xyc})^{\alpha}}$ allow to discount the loss for easy positives and negatives in order to let the network focus on hard to classify examples.

3. $\boldsymbol{(1-y_{xyc})^{\beta}}$ reduce the penalization of the loss at positions near the ground truth (negative positions) allowing the network to classify them as centers.

2.2 Algorithms

2.2.1 IAN detection with template matching guided by Ariadne

The core idea is to track the IAN position frame by frame producing different paths. The one with the best curvature is selected. In order to do so at each step, patches at previous positions are extracted from the previous frame and a scores map (Figure 2.3c) is computed sliding each patch (Figure 2.3a) onto



(a) Patch extracted on the previous frame.

(b) RoI considered at current frame.

(c) Scores map computed.

(d) $k_1 = 10$ points selected using the scores map. Then $k_2 = 5$ points are selected by Ariadne (blue border).

Figure 2.3: Template matching applied considering on of the previous point.

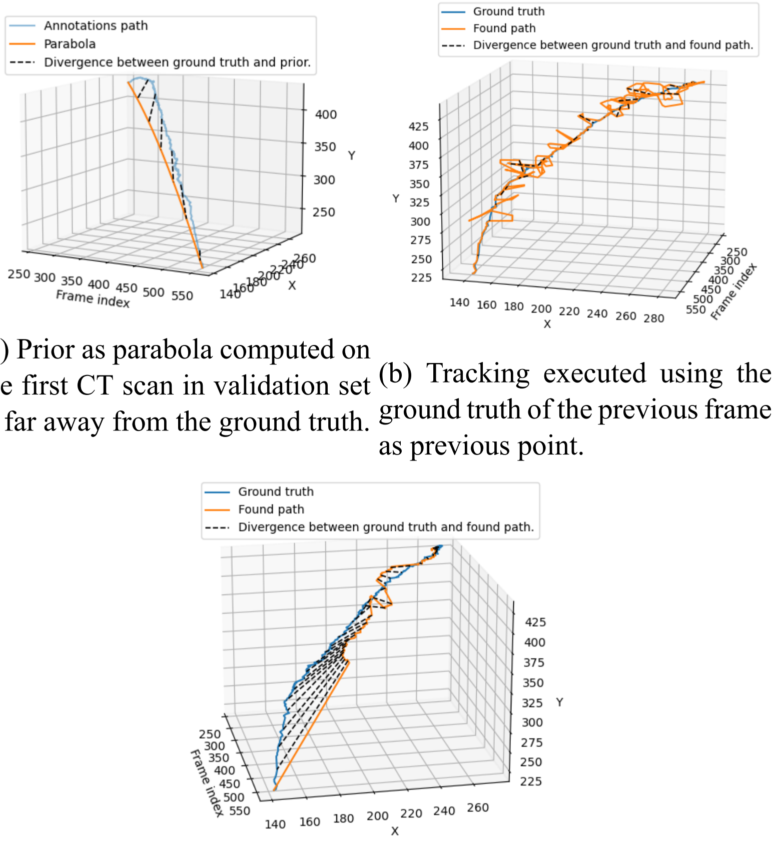
a region of interest (RoI) (Figure 2.3b) centered on the previous position at the current frame and computing a score using the *Normalized Cross Correlation (NCC)*. The NCC is the following:

$$NCC(x,y) = \sum_{t_x=0}^{w} \sum_{t_y=0}^{h} \frac{T(t_x,t_y) * I(x+t_x, y+t_y)}{\sqrt{\sum_{t_x=0}^{w} \sum_{t_y=0}^{h} T(t_x,t_y)^2 \sum_{t_x=0}^{w} \sum_{t_y=0}^{h} I(x+t_x, y+t_y)^2}}$$

where I is the image (the RoI in our case), T is the patch and w and h are the width and height of the patch. This technique is called **template matching**. Then, these points are filtered with the statistical reasoning done by Ariadne for acquiring possible candidates.

In particular, a single step of the algorithm compute k_1 points using template matching; although in Ariadne it's selected the point which maximize the probability of a path (see Equation 2.1), we select k_2 points with the highest probability considering only the Distance and Curvature likelihood (Figure 2.3d).

Since this is a *breadth first search* algorithm, in order to prune invalid paths we thought to discard points far away from a precomputed pripor modelled as a **parabola** but since it wasn't robust (we will discuss this in Section 3.4.1), this pruning was avoided. We observed that most of the uncertainty was in the middle, hence we allowed new paths only in the center. In particular given a probability sampled from a normal distribution $\mathcal{N}(\left\lfloor \frac{L}{2} \right\rfloor, \sigma^2)$ where L is equal to the amount of frames and σ^2 represent the amount of frames to consider in the middle we chose to insert in the frontier only points whose probability weighted by the Gaussian was greater than a threshold. Moreover, in order to avoid an exponential number of paths, an upper bound of possible opened path was set.



(a) Prior as parabola computed on the first CT scan in validation set is far away from the ground truth.

(b) Tracking executed using the ground truth of the previous frame as previous point.

(c) Tracking executed using detection of previous frame as previous point.

Figure 2.4: Parabola and tracking done on a CT scan in the validation set showing each position in a 3-dimensional space x-y-frame.

2.2.2 IAN detection with CenterNet

As in Section 2.2.1, we are going to track IAN positions: in particular CenterNet detections will be used in order to obtain them, exploiting spatial and temporal information between consecutive frames, that is IAN positions should not be far away from each other and positions detected starting from both sides at same frame should be closer.

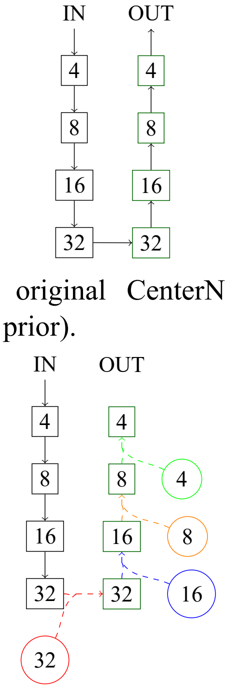
The former is ensured reweighting the heatmap generated by CenterNet with a prior, the latter is done, simply, computing at each frame the mean between detections found starting from the beginning and the ending of the nerve.

For our task CenterNet was modified for predicting a heatmap $H \in R^{\frac{H}{D}\times\frac{W}{D}\times 1}$ since there are no classes, a radius map $R \in R^{\frac{H}{D}\times\frac{W}{D}\times 1}$ other than an offset map.

We defined the **prior** P_i at step i as a Gaussian centered on the previous position p_{i-1} whose scale and sigma are hyperparameters: **prior scale** A_{prior} and **prior sigma** σ_{prior}:

$$P_i = A_{prior} * e^{\frac{(x-p_{i-1,x})^2+(y-p_{i-1,y})^2}{2\sigma^2_{prior}}}$$

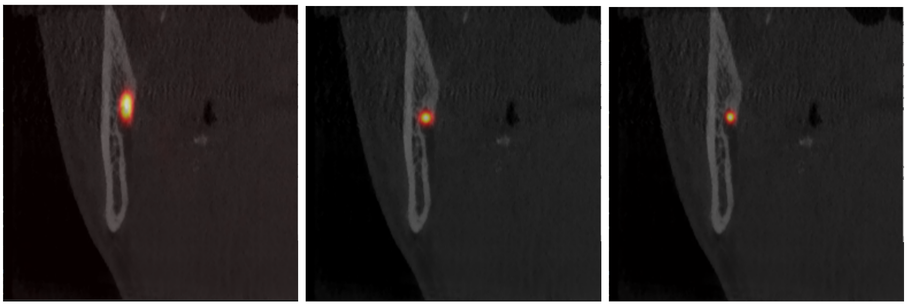
Therefore the **tracking** is done in the following way: at each step i, given the CenterNet's prediction $\hat{H}_i$(Figure 2.5a) and the prior P_i (Figure 2.5b) we compute a combined heatmap $\text{HP}_i = \hat{H}_i * P_i$ (Figure 2.5c). Then onto a RoI centered on the previous position p_{i-1} we apply NMS as in the original paper and extract the point p_i with the greatest score.



(a) The original CenterNet model (without prior).

(b) CenterNet model using prior injected at four possible levels.

Figure 2.6: Models used in our work. The numbers represent the downscale used by the original paper.



(a) Heatmap generated by CenterNet

(b) Prior heatmap generate at previous position

(c) Heatmap composed by the CenterNet and prior one.

Figure 2.5: Heatmap generated during one step of the tracking.

The algorithm is applied in both directions, from the mental foramen to Spix spine and vice versa, producing two set of positions P_f and P_b that are used for computing the averaged ones: given $p_f \in P_f$ and $p_b \in P_b$, the final position $p_m = \frac{p_f + p_b}{2}$.

Level	Downscale factor in the decoder (Figure 2.6b)
4	32
3	16
2	8
1	4

Table 2.1: Levels associated to each downscale factor

Moreover, we tried to track the position injecting the combined heatmap found before in the model, trying to exploit this information during the inference. In order to do so we define 4 possible levels along the network before each deformable convolution (Figure 2.2) indexing them from 4 to 1 starting from the deepest location to the shallowest one (Table 2.1). The heatmap HP_i is downscaled for complying with the spatial resolution used by the original network and then concatenated along the channel with the features map computed by the previous convolution (Figure 2.6b).

Regularization when using a prior When injecting a prior inside CenterNet we observed **heavy overfitting**: not only was the heatmap generated depending on the prior, that is, prior and heatmap were very similar, but also its confidence was lower, the lower the scale A_{prior}. In Figure 2.7 CenterNet receives in input an image with random values sampled from a standard normal distribution $I \sim N(0, 1)$ and a prior centered at the center of the image with different σ_{prior}: when $A_{prior} = 0.3$ the heatmap generated by the model without dropout presents more uncertainty than when using $A_{prior} = 1.0$, where no uncertainty is present. Hence during the training we applied the following regularizations:

- Given the downscale factor of the prior d_{prior} at level where is injected (Table 2.1), we distorted the center of the prior with noise sampled from a multivariate normal distribution $\mathcal{N}(0, \sigma^2)$ with sigma equals to 10, in order to have in the full resolution an offset equals to $\pm d_{prior} \times 3 \times \sqrt{10}$ forcing the network to learn to predict the correct heatmap: this should reflect the real situation during the tracking where the prior P_i position is not perfect.

- Then we also dropped out randomly, with a given probability $p_{dropout}$, the prior within the batch replacing it with a zeroed tensor.

The effect of these regularizations can be seen in Figure 2.7 where the model with dropout trained with priors whose position was distorted by an offset sampled from $\mathcal{N}(0, \sigma^2)$, doesn't predict the prior.

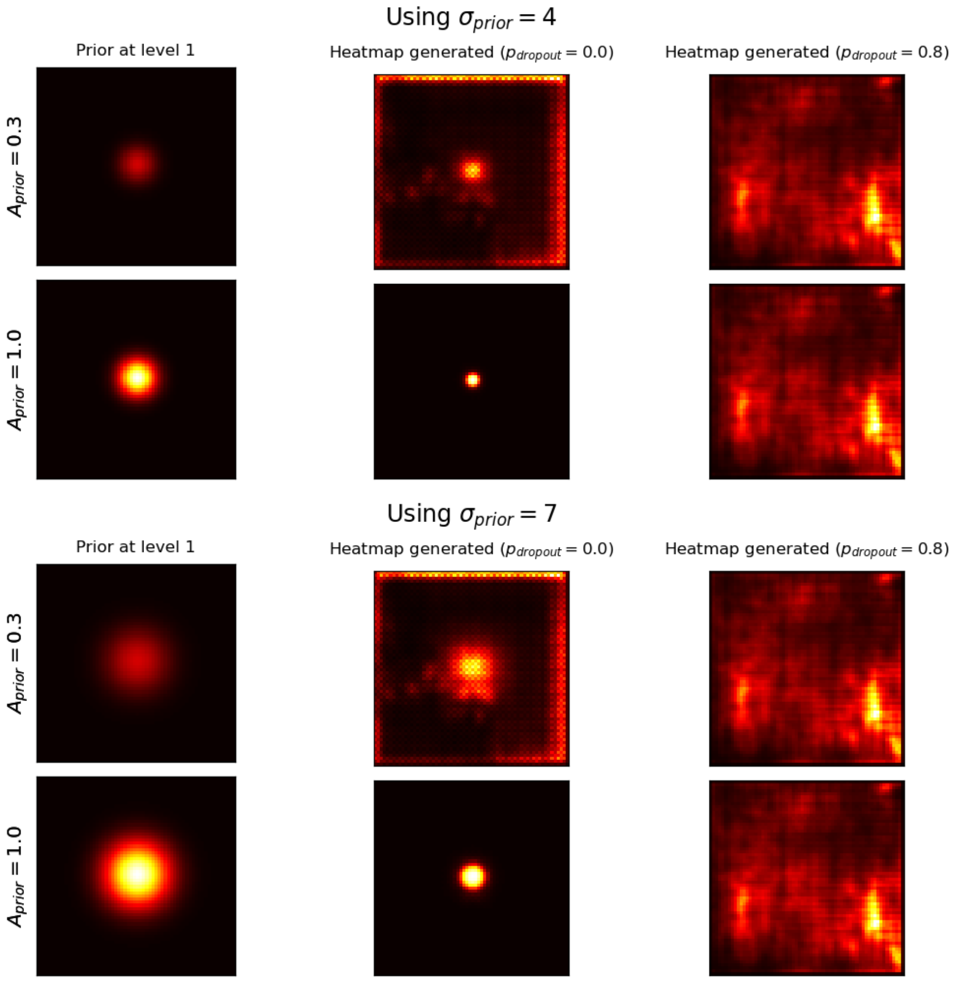


Figure 2.7: Heatmap generated by models trained with and without dropout, using the same random input with prior injected at level 1 and different scales and sigmas.

Chapter 3

Experiments

3.1 Dataset

The entire dataset is composed by $\sim$ 71 (CB)CT scan **anonymized**, in order to respect GDPR gudelines [1], annotated by a **non expert** (myself).

DICOM data are processed using *pydicom*[23]: *raw value* are converted into the *Housenfield units* thorugh a linear conversion $y = m * x + c$ where m and c are repsectively *RescaleSlope* and *RescaleIntercept*, parameters of the machine that created the data. Moreover in order to visualize particular information (tissues, bones, ...), *windowing*[2] is applied, that is, given the *WindowWidth*(WW) and *WindowCenter*(WL), the Housenfield information is clipped between WL ± $\frac{\text{WW}}{2}$ where the lowest value is set to black and the

[1] https://gdpr-info.eu/recitals/no-26/

[2] https://dicom.nema.org/medical/dicom/current/output/chtml/part03/sect_C.11.2.html#sect_C.11.2.1.2

highest value is set to white (due to MONOCHROME2 flag [3]):

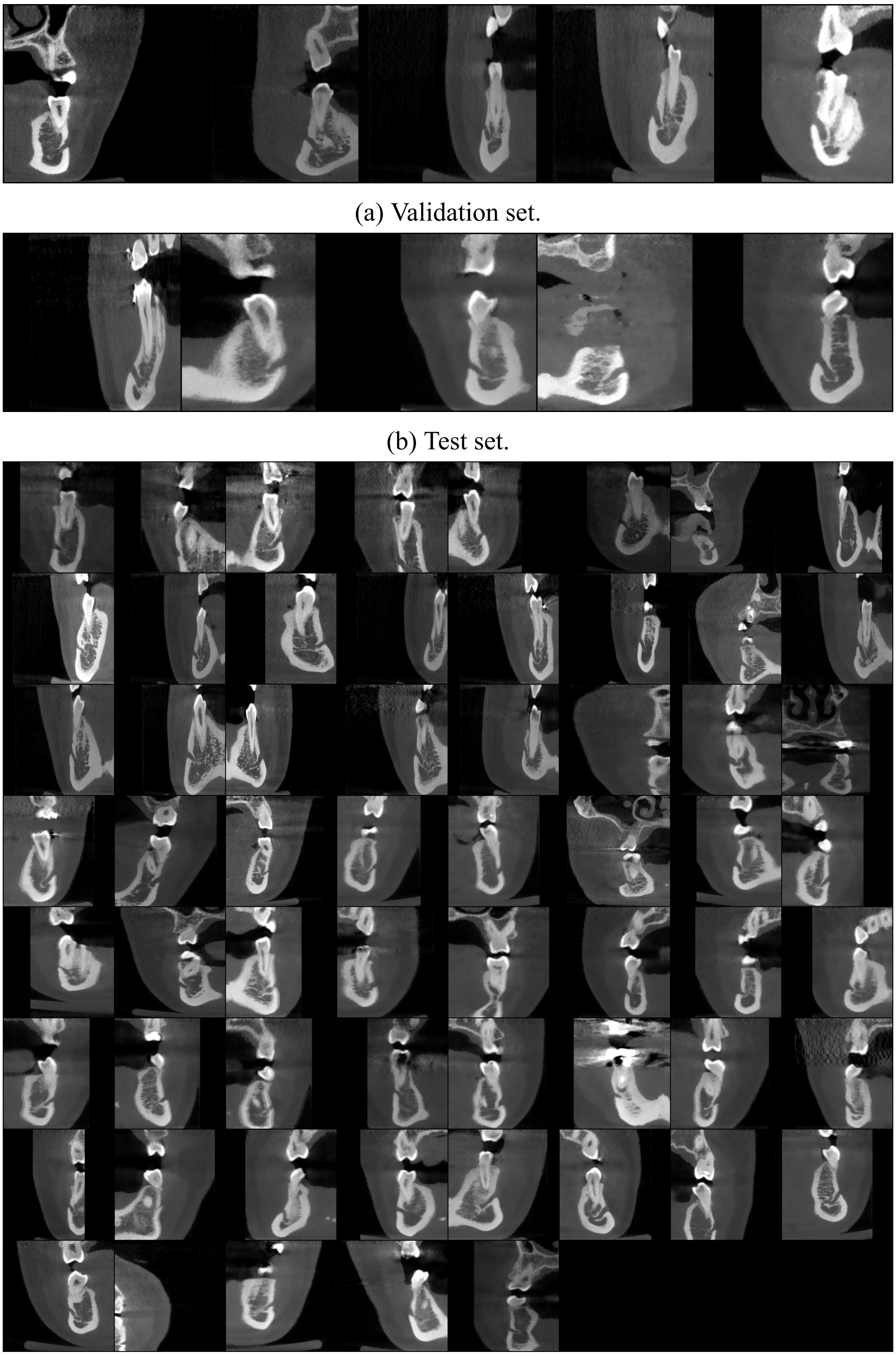
$$p(h) = \begin{cases} 0, & h \leq \text{WL} - \frac{\text{WW}}{2} \\ \left(\frac{h-(\text{WL}-0.5)}{\text{WW}-1} + 0.5\right) * 255, & \text{WL} - \frac{\text{WW}}{2} \leq h < \text{WL} + \frac{\text{WW}}{2} \\ 255, & h > \text{WL} + \frac{\text{WW}}{2} \end{cases} \tag{3.1}$$

where h is the Housenfield value and $p(h)$ is the pixel value associated computed from the Housenfield one.

When the IAN is well visible, annotation are inserted at the center otherwise hints are used in order to put it in the right position as much as possibile. The Figure 3.1 contains the whole dataset used for training, validating and testing the network.

Table 3.1 contains a comparison between our dataset and ones used by papers cited in Section 1.1: not only the amount of data used for training is much larger than ours, but each scan was annotated by radiographer with voxel-level precision. This highlight another important aspect of our work: the annotations required are much easier to obtain than ones used in others since it is simpler to annotate the center of the canal with a circle than annotating pixels in a volume.

[3] https://dicom.nema.org/medical/dicom/current/output/chtml/part03/sect_C.7.6.3.html



(a) Validation set.

(b) Test set.

(c) Training set.

Figure 3.1: Mental foramen cut for the whole dataset.

Source	Train	Val	Test
This	61	5	5
Kainmueller et al.	106		
Jaskari et al.	637	52	15[4] 128[5]
Liu et al.	154	30	45
Bayrakdar et al. (75 CT scans)			
Kwak et al.	~ 29456 (images)	~ 9818 (images)	~ 9818 (images)

Table 3.1: Dataset comparison (CT scans).

3.2 Metrics

3.2.1 Intersection over Union (IoU)

The Intersection over Union is a metric used in object detection for addressing how much two bounding boxes overlap. This metric is scale invariant.

Given the ground truth bounding box (B_g) and detection bounding box (B_d), the IoU is:

$$IoU = \frac{|B_g \cap B_d|}{|B_g \cup B_d|}$$

with $IoU \in [0, 1]$. In practice, it is the ratio between the pixels belonging to both ground truth and detection over the pixels belonging to either the ground truth or the detection.

[4] Primary test data

[5] Secondary test data

3.2.2 Generalized Intersection over Union (GIoU)

When two bounding boxes are not overlapping the IoU is equal to 0: no insight is given about much far away they are. Therefore, GIoU was proposed [27]:

$$GIoU = \frac{|B_g \cap B_d|}{|B_g \cup B_d|} - \frac{|C \setminus (B_g \cup B_d)|}{|C|}$$

where C represents the minimum convex hull that encloses the two bounding boxes and $GIoU \in [-1, 1]$. Moreover, GIoU is a differentiable IoU therefore can be used as a loss.

3.2.3 Distances in millimeters

A CT scan volume is composed by slices of the head from top to bottom. This is the **axial** plane. On an axial plane we know the *row* and *column spacing*, that are, respectively, the vertical and horizontal distance between pixel centers in millimeters. Then the distance in millimeters between two slices is the *slice thickness*.

Therefore in order to translate distances from pixels to millimeters in the **coronal plane** we need the *slice thickness* and *column spacing* (Figure 3.2). Given the points p_1 and p_2, the *slice thickness* r_y and the *column spacing* r_x the computation is the following:

$$d_{mm}(p_1, p_2) = \sqrt{(r_x(p_{2,x} - p_{1,x}))^2 + (r_y(p_{2,y} - p_{1,y}))^2} \tag{3.2}$$

since in our case the *slice thickness* and the *column spacing* were the same, thus $r_x = r_y = r$, the formulation is simplified as:

$$d_{mm}(p_1, p_2) = r * ||p_2 - p_1||_2 \tag{3.3}$$

where $||\cdot||$ is the Euclidean distance.

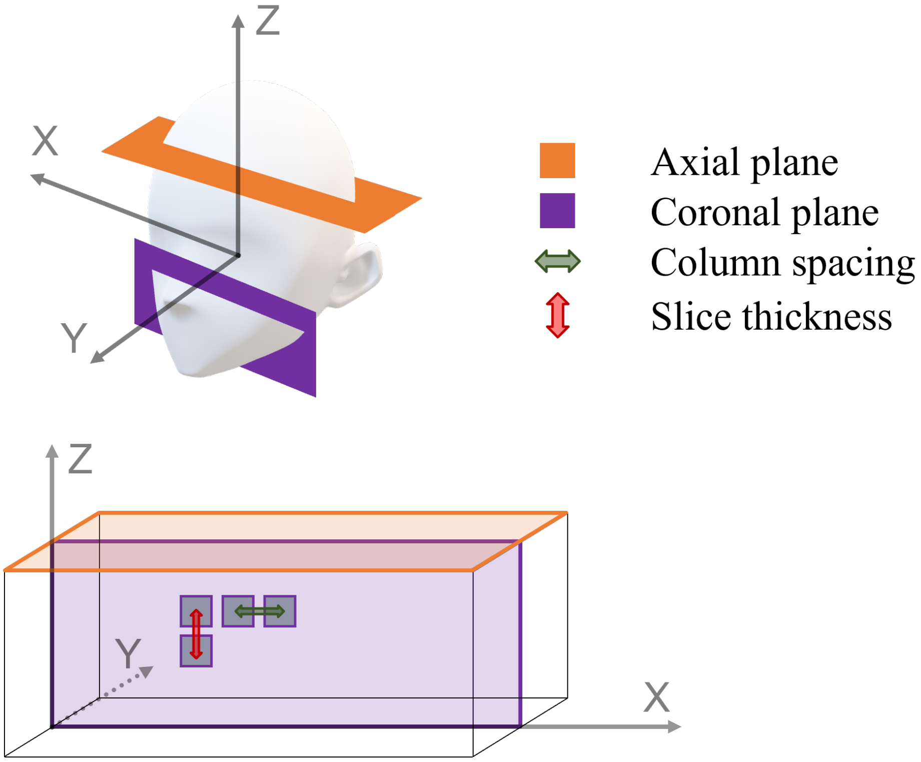


Figure 3.2: CT scan volume where are presented the column spacing and slice thickness.

3.3 Training and validation

Training CenterNet (trained on COCO dataset) was finetuned on 61 CT scans (hence $\sim$ 122 canals, since for each CT scan we have the left and right one) for 400 epochs with a learning rate equals to $5 * 10^{-4}$ using Adam with default parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$). Moreover, following the paper, we set the bias of the heatmap head at -9.21 using

$$b = -\log(\frac{1-\pi}{\pi}) \tag{3.4}$$

where π is the prior probability of the foreground class. In our case, given $\pi = 2 * 10^{-4}$ (since there is always one pixel as ground truth in the heatmap at size 64×64), $b = -9.21$, such that it will predict at the beginning of the training $\sigma(-9.21) = 10^{-4}$. Indeed, we tried -2.19, -4.59, -6.9 along

-9.21 which gives us the lowest initial loss.

Each frame was resized to 256×256 and then downscaled by a factor 4 (following the original paper). In order to increase the amount of possible data seen by the network, the dataset was augmented with *zoom in/zoom out* and *horizontal flipping* randomly applied at each frame (Table 3.2 contains hyperparameters values).

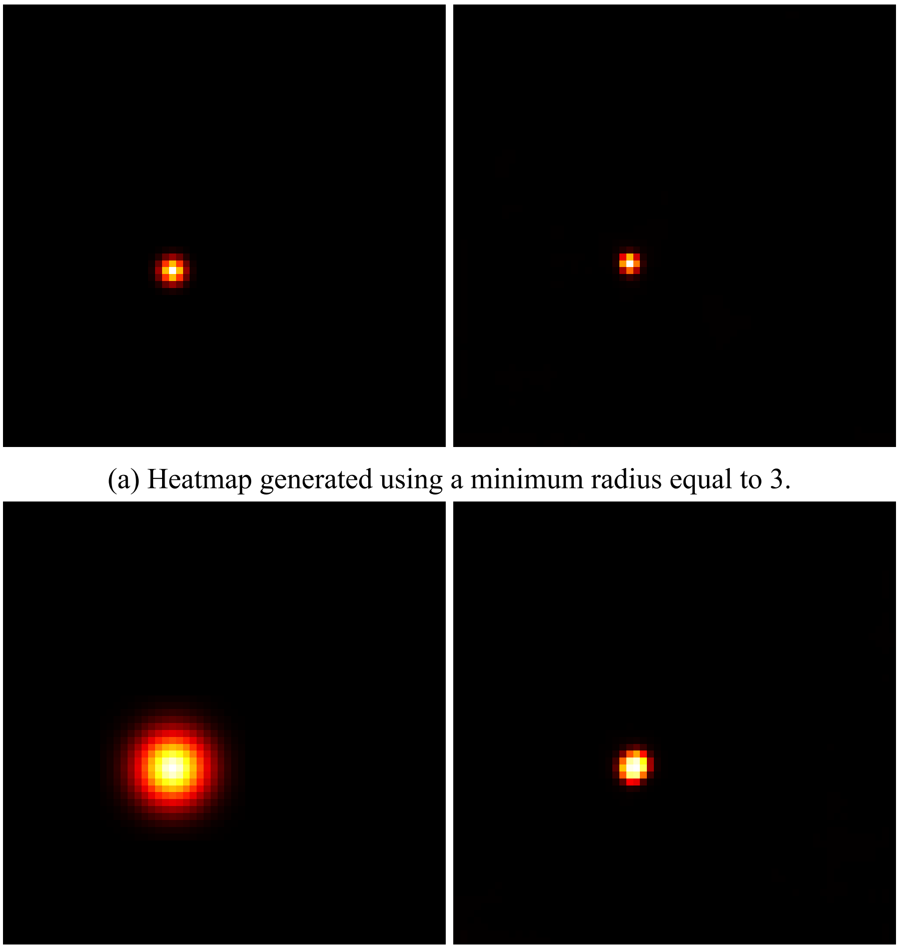
Due to smaller annotation radius values after rescaling, a minimum value was enforced to the σ_p on Equation (2.7) used for generating the heatmap (equal to 10), which allowed to obtain less restrictive output heatmaps from CenterNet (Figure 3.3).

In order to ease the problem, for all experiments we used a fixed radius equals to the mean radius computed, at resolution 256×256, over the training set ($r = 7$). We would like to highlight that the radius is not really important for our study since we are more interested in the center position, and so in the distance in millimeters from the real IAN, than in the width of the canal. Nevertheless, the network can be trained also for predicting it.

Validation During the training every 5 epochs we executed the tracking, using default parameters (Table 3.3), for measuring IoU, GIoU and distance in *mm* between the ground truth and mean detections in order to have meaningful metrics. The ones with the greatest GIoU were selected.

Name	Value
Probability horizontal flip	0.8
Probability zoom in/out	0.8
Zoom in/out range	$0.8 - 1.5$
$p_{dropout}$	0.8

Table 3.2: Augmentation hyperparameters.



(a) Heatmap generated using a minimum radius equal to 3.

(b) Heatmap generated using a minimum radius equal to 10.

Figure 3.3: Ground truth heatmap (left) and heatmap generated by CenterNet (right)

3.4 Results

3.4.1 IAN detection with template matching guided by Ariadne

Since the time complexity of the algorithm increase exponentially with the number of k_2 paths, the evaluation was done using only the first CT scan in the validation set (Figure 3.1a).

The prior as parabola is not robust As mentioned in Section 2.2.1 the precomputed prior as parabola wasn't robust enough. It was created computing parameters of a generic parabola using the *least square solution*. In particular, we normalize the x and y coordinates over width and height of the start, middle and ending annotations for each canal. Then given the matrix

$$\begin{bmatrix} x_1^2 & x_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n^2 & x_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} y_1 \\ \dots \\ y_n \end{bmatrix} \tag{3.5}$$

,that is $XA = Y$, we solve for $A = (X^T X)^{-1}(X^T Y)$. The approximation (Figure 2.4a) is far away from the ground truth therefore can't be used as prior.

The tracking diverges Another problem encountered that undermine the whole algorithm is the divergence of the tracking caused by the wrong template extracted in the previous frame. This can be observed comparing Figure 2.4b and Figure 2.4c: enforcing the position, allow the algorithm to correct itself as soon as it diverges from the right path.

3.4.2 IAN detection with CenterNet

Our experiments were evaluated computing IoU, GIoU and distances in millimeters between ground truth and detection obtained either by using tracking described in Section 2.2.2 or by CenterNet alone. For the tracking variant or our method, we used detections from start to end (**Backward**), end to start (**Forward**) and averaged ones (**Mean**). For a fair comparison default hyperparameters (Table 3.3) were used for all experiments. The results are displayed in **Table 3.4** and **Table 3.5**.

In figure Figure 3.8 we show metrics and the tracking computed on one of the best CT scan in our test set.

Name	Value
A_{prior}	1.0
σ_{prior}	$(4,4)$

Table 3.3: Tracking hyperparameters used for validating and testing our models.

CenterNet on its own is not accurate enough Despite using **ResNet101** as backbone results don't improve with respect using **ResNet18**. This is expected due to low amount of data used for training (Table 3.1). Moreover **CT scans are unbalanced**: the amount of frames where Spix spine (Figure 1.1a) and mental foramen (Figure 1.1b) annotation are present, is much lower than ones where the IAN is in the body of the pathway: this lead to uncertainty at the start and end of the canal (Figure 3.8b). Also noise within (CB)CT scan lower model performances.

The prior can increase the performances A slightly improvements can be seen despite it has no real consequences when comparing distances in millimeters (Table 3.4).

The prior has to be injected at deeper level Better performances are registered when the prior is inserted at deeper level but again it has no real impact on real distances (Figure 3.5). A possible explanation could be that due to more non-linearities, the network is able to extract more information in order to infer positions.

Void bones are dangerous CenterNet is unable to detect IAN when it is hidden by void bones as in Figure 3.7, therefore tracking will fail unless CenterNet is able to recover the correct position.

Tracking allows mitigating bad detections Despite wrong detections from CenterNet, tracking allows partially to correct them (Figure 3.8b) and produces smoother trajectories.

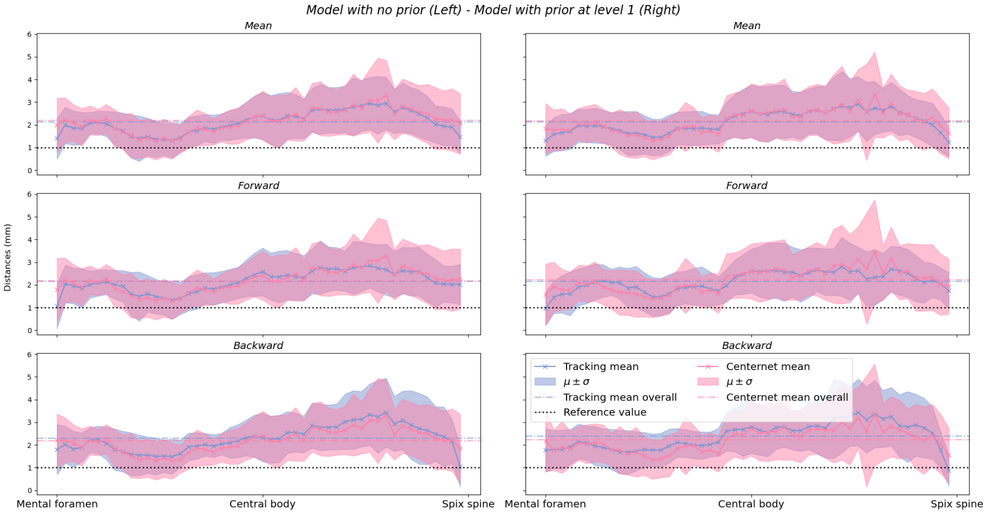


Figure 3.4: Evaluation of the model without and with prior at level 1.

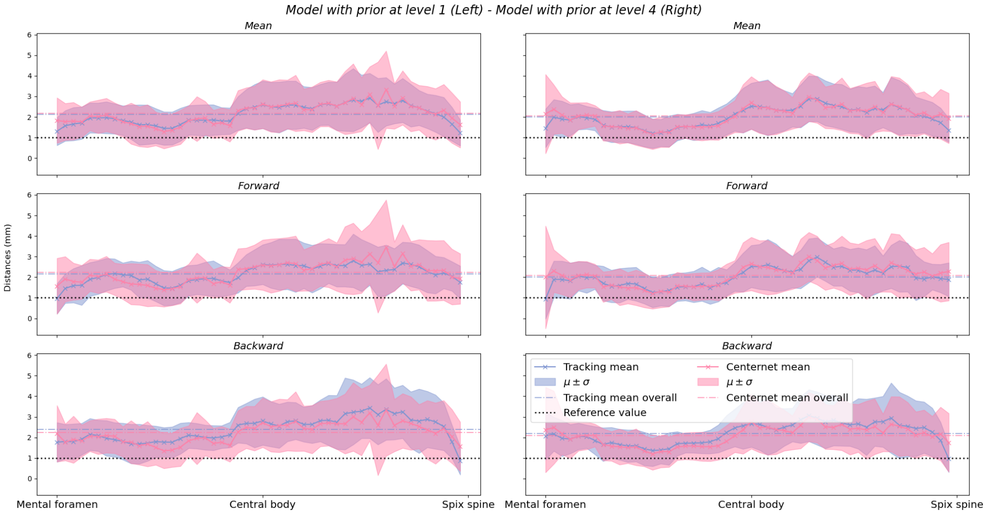


Figure 3.5: Evaluation of the model with prior at level 1 and 4.

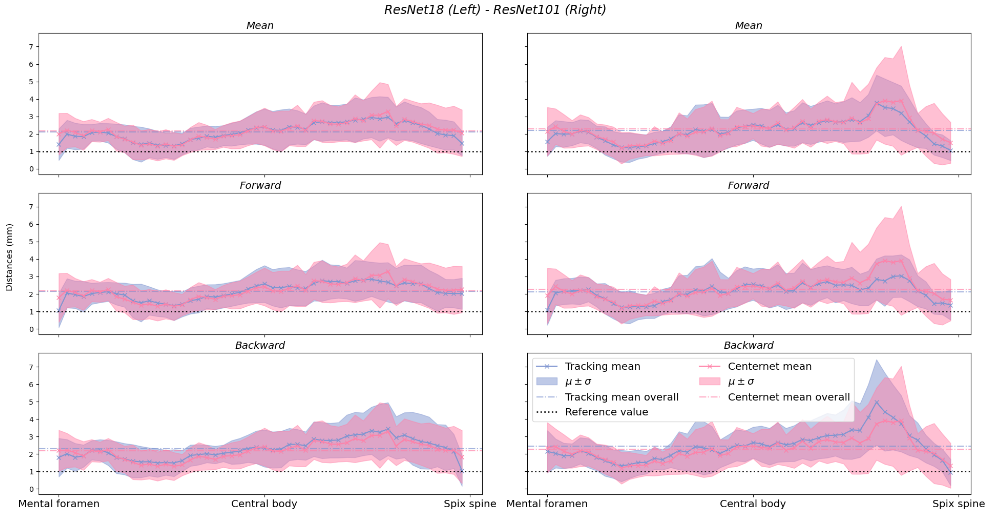


Figure 3.6: Evaluation of the ResNet18 and ResNet101 as backbone with no prior.

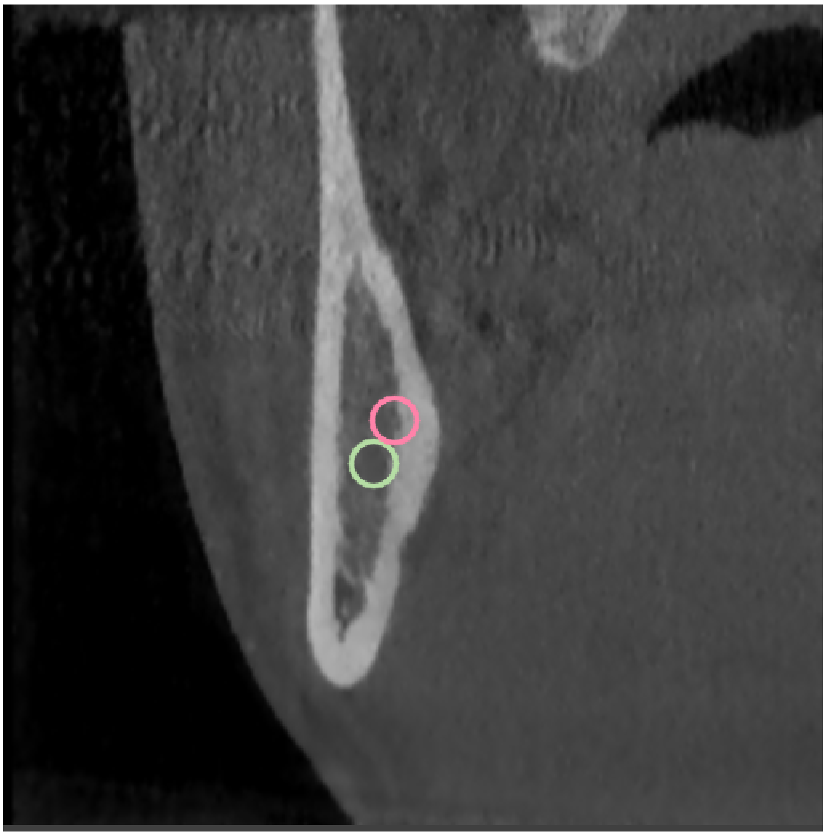
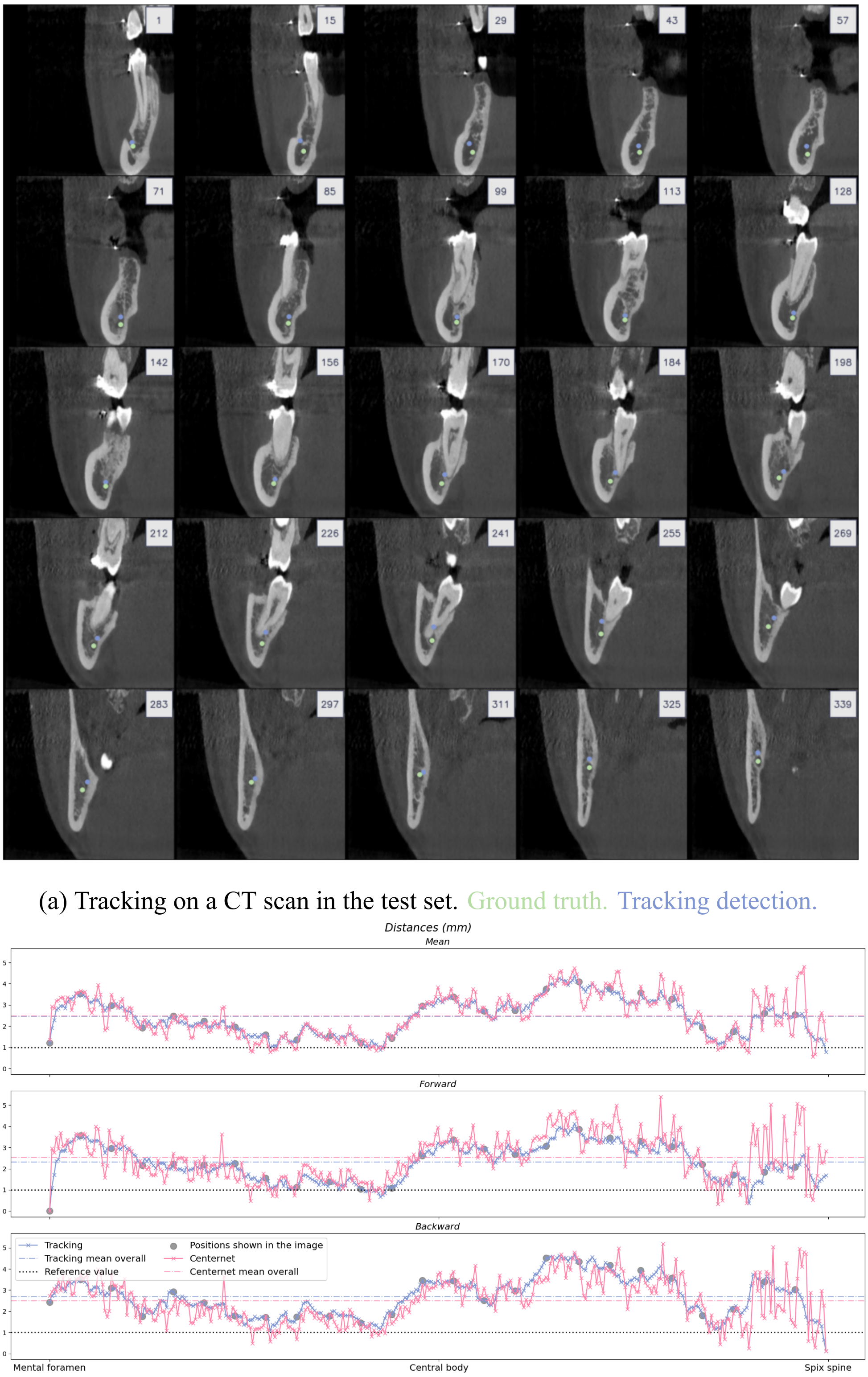


Figure 3.7: Frame from a CT scan in test set where IAN is hidden. Ground truth. CenterNet detection.



(a) Tracking on a CT scan in the test set. Ground truth. Tracking detection.

(b) Metrics on a CT scan in the test set.

Figure 3.8: Detection obtained with ResNet18 and injecting the prior at level 4.

Model		Mean		
Backbone	Configuration	IoU ↑	GIoU ↑	Distances(mm) ↓
ResNet18	P0 (Tracking)	0.236 ± 0.206	0.081 ± 0.317	2.116 ± 1.007
	P0 (CenterNet)	0.232 ± 0.213	0.069 ± 0.327	2.182 ± 1.141
ResNet101	P0 (Tracking)	0.241 ± 0.218	0.080 ± 0.340	2.181 ± 1.197
	P0 (CenterNet)	0.239 ± 0.221	0.070 ± 0.350	2.274 ± 1.476
ResNet18	P1 (Tracking)	0.232 ± 0.218	0.082 ± 0.326	2.131 ± 1.060
	P1 (CenterNet)	0.236 ± 0.226	0.079 ± 0.342	2.171 ± 1.208
ResNet18	P4 (Tracking)	**0.259 ± 0.215**	**0.119 ± 0.323**	**2.004 ± 1.017**
	P4 (CenterNet)	0.254 ± 0.219	0.106 ± 0.330	2.051 ± 1.104

Table 3.4: Metrics computed using averaged detections.

Model		Forward		
Backbone	Configuration	IoU ↑	GIoU ↑	Distances(mm) ↓
ResNet18	P0 (Tracking)	0.233 ± 0.201	0.084 ± 0.300	2.147 ± 1.007
	P0 (CenterNet)	0.230 ± 0.212	0.066 ± 0.326	2.182 ± 1.147
ResNet101	P0 (Tracking)	0.246 ± 0.219	0.097 ± 0.325	2.112 ± 1.098
	P0 (CenterNet)	0.237 ± 0.219	0.067 ± 0.349	2.272 ± 1.479
ResNet18	P1 (Tracking)	0.227 ± 0.206	0.078 ± 0.305	2.147 ± 1.024
	P1 (CenterNet)	0.229 ± 0.223	0.070 ± 0.341	2.221 ± 1.295
ResNet18	P4 (Tracking)	0.259 ± 0.213	0.123 ± 0.313	2.014 ± 1.000
	P4 (CenterNet)	0.249 ± 0.221	0.097 ± 0.334	2.088 ± 1.174

Model		Backward		
Backbone	Configuration	IoU ↑	GIoU ↑	Distances(mm) ↓
ResNet18	P0 (Tracking)	0.214 ± 0.206	0.046 ± 0.319	2.286 ± 1.139
	P0 (CenterNet)	0.230 ± 0.212	0.066 ± 0.326	2.182 ± 1.146
ResNet101	P0 (Tracking)	0.209 ± 0.198	0.028 ± 0.329	2.451 ± 1.409
	P0 (CenterNet)	0.237 ± 0.220	0.066 ± 0.349	2.275 ± 1.481
ResNet18	P1 (Tracking)	0.184 ± 0.198	0.012 ± 0.311	2.385 ± 1.188
	P1 (CenterNet)	0.227 ± 0.222	0.065 ± 0.342	2.228 ± 1.285
ResNet18	P4 (Tracking)	0.217 ± 0.201	0.058 ± 0.309	2.189 ± 1.127
	P4 (CenterNet)	0.249 ± 0.222	0.097 ± 0.335	2.089 ± 1.162

Table 3.5: Metrics computed using only detection from end-to-start (Forward) and start-to-end (Backward).

Chapter 4

Conclusion

The inferior alveolar nerve (IAN) is an important nerve whose position have to be known with millimetric precision in order to avoid damaging it during surgical operations or implants positioning.

In this thesis, we explored two approaches to automatically identify it in CT scans: tracking of IAN using template matching detections with statistical reasoning (inspired by Ariadne [11]); a novel way to address this problem trough tracking its position frame by frame using CenterNet[33] detections. In the latter approach, we also explored the idea of enforcing spatial and temporal constraints to link detections across consecutive CT scans. Moreover, we tried to exploit the same spatial information within the network in order to influence its prediction.

The approach based on Ariadne was not robust enough and tracking diverges after a few frames. The method based on CenterNet and tracking information was able to achieve good performance for a preliminary study. The best configuration predicts the position with an average error of 2 mm.

Different ablation studies show the effectiveness of tracking using a prior and injecting the prior inside the network, in particular at the deeper level.

A possible continuation of this study could explore different ways to leverage the prior information as well as different networks: for instance, due to the sequentiality of the data Recurrent CNN or 3D CNN could achieve better performances.

Moreover increasing the quantity and enhancing the quality of the dataset will for sure increase the ability of each model to perform better also in edge case situations improving overall the performances.

Finally due to the sensible application, explainability of the model prediction should be considered, in order to let the clinic user makes a better informed decision instead of one based blindly on the final inference.

Bibliography

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012. DOI: 10.1109/TPAMI.2012.120.

[2] J. O. Agbaje, E. V. de Casteele, A. S. Salem, D. Anumendem, I. Lambrichts, and C. Politis. Tracking of the inferior alveolar nerve: its implication in surgical planning. *Clinical Oral Investigations*, 21(7):2213–2220, November 2016. DOI: 10.1007/s00784-016-2014-x. URL: https://doi.org/10.1007/s00784-016-2014-x.

[3] A. A. Alhassani and A. S. T. AlGhamdi. Inferior alveolar nerve injury in implant dentistry: diagnosis, causes, prevention, and management. *Journal of Oral Implantology*, 36(5):401–407, October 2010. DOI: 10.1563/aaid-joi-d-09-00059. URL: https://doi.org/10.1563/aaid-joi-d-09-00059.

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561, 2015. arXiv: 1511.00561. URL: http://arxiv.org/abs/1511.00561.

[5] S. K. Bayrakdar, K. Orhan, I. S. Bayrakdar, E. Bilgir, M. Ezhov, M. Gusarev, and E. Shumilov. A deep learning approach for dental

implant planning in cone-beam computed tomography images. *BMC Medical Imaging*, 21(1), May 2021. DOI: 10.1186/s12880-021-00618-z. URL: https://doi.org/10.1186/s12880-021-00618-z.

[6] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. *CoRR*, abs/1606.06650, 2016. arXiv: 1606.06650. URL: http://arxiv.org/abs/1606.06650.

[7] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. *CoRR*, abs/1703.06211, 2017. arXiv: 1703.06211. URL: http://arxiv.org/abs/1703.06211.

[8] Z. Dai, H. Liu, Q. V. Le, and M. Tan. Coatnet: marrying convolution and attention for all data sizes, 2021. arXiv: 2106.04803 [cs.CV].

[9] R. Girshick. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015. DOI: 10.1109/ICCV.2015.169.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):142–158, 2016. DOI: 10.1109/TPAMI.2015.2437384.

[11] D. D. Gregorio, G. Palli, and L. di Stefano. Let's take a walk on superpixels graphs: deformable linear objects segmentation and model estimation. *CoRR*, abs/1810.04461, 2018. arXiv: 1810.04461. URL: http://arxiv.org/abs/1810.04461.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[13] J. Iwanaga, Y. Matsushita, T. Decater, S. Ibaragi, and R. S. Tubbs. Mandibular canal vs. inferior alveolar canal: evidence-based terminology analysis. *Clinical Anatomy*, 34(2):209–217, August 2020. DOI: `10.1002/ca.23648`. URL: `https://doi.org/10.1002/ca.23648`.

[14] J. Jaskari, J. Sahlsten, J. Järnstedt, H. Mehtonen, K. Karhu, O. Sundqvist, A. Hietanen, V. Varjonen, V. Mattila, and K. Kaski. Deep learning method for mandibular canal segmentation in dental cone beam computed tomography volumes. *Scientific Reports*, 10(1), April 2020. DOI: `10.1038/s41598-020-62321-3`. URL: `https://doi.org/10.1038/s41598-020-62321-3`.

[15] D. Kainmueller, H. Lamecker, H. Seim, M. Zinser, and S. Zachow. Automatic extraction of mandibular nerve and bone from cone-beam CT data. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009*, pages 76–83. Springer Berlin Heidelberg, 2009. DOI: `10.1007/978-3-642-04271-3_10`. URL: `https://doi.org/10.1007/978-3-642-04271-3_10`.

[16] K. S. Krishnan and K. S. Krishnan. Vision transformer based covid-19 detection using chest x-rays. *2021 6th International Conference on Signal Processing, Computing and Control (ISPCC)*, October 2021. DOI: `10.1109/ispcc53510.2021.9609375`. URL: `http://dx.doi.org/10.1109/ISPCC53510.2021.9609375`.

[17] G. H. Kwak, E.-J. Kwak, J. M. Song, H. R. Park, Y.-H. Jung, B.-H. Cho, P. Hui, and J. J. Hwang. Automatic mandibular canal detection using a deep convolutional neural network. *Scientific Reports*, 10(1), March 2020. DOI: `10.1038/s41598-020-62586-8`. URL: `https://doi.org/10.1038/s41598-020-62586-8`.

[18] J. Li, D. Cai, and X. He. Learning graph-level representation for drug discovery, 2017. arXiv: `1709.03741 [cs.LG]`.

[19] Y. Li, Y. Chen, N. Wang, and Z. Zhang. Scale-aware trident networks for object detection, 2019. arXiv: 1901.01892 [cs.CV].

[20] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dolla'r. Focal loss for dense object detection. *CoRR*, abs/1708.02002, 2017. arXiv: 1708.02002. URL: http://arxiv.org/abs/1708.02002.

[21] M.-Q. Liu, Z.-N. Xu, W.-Y. Mao, Y. Li, X.-H. Zhang, H.-L. Bai, P. Ding, and K.-Y. Fu. Deep learning-based evaluation of the relationship between mandibular third molar and mandibular canal on CBCT. *Clinical Oral Investigations*, 26(1):981–991, July 2021. DOI: 10.1007/s00784-021-04082-5. URL: https://doi.org/10.1007/s00784-021-04082-5.

[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: single shot multibox detector. *Lecture Notes in Computer Science*:21–37, 2016. ISSN: 1611-3349. DOI: 10.1007/978-3-319-46448-0_2. URL: http://dx.doi.org/10.1007/978-3-319-46448-0_2.

[23] D. Mason and et al. Pydicom/pydicom: pydicom 2.2.2, version v2.2.2, October 2021. DOI: 10.5281/zenodo.5543955. URL: https://doi.org/10.5281/zenodo.5543955.

[24] K. Nazeri, H. Thasarathan, and M. Ebrahimi. Edge-informed single image super-resolution, 2019. arXiv: 1909.05305 [eess.IV].

[25] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2017. DOI: 10.1109/CVPR.2017.690.

[26] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017. DOI: 10.1109/TPAMI.2016.2577031.

[27] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. Generalized intersection over union, June 2019.

[28] O. Ronneberger, P. Fischer, and T. Brox. U-net: convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. arXiv: `1505.04597`. URL: `http://arxiv.org/abs/1505.04597`.

[29] J. Rood and B. N. Shehab. The radiological prediction of inferior alveolar nerve injury during third molar surgery. *British Journal of Oral and Maxillofacial Surgery*, 28(1):20–25, February 1990. DOI: `10.1016/0266-4356(90)90005-6`. URL: `https://doi.org/10.1016/0266-4356(90)90005-6`.

[30] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1605.06211, 2016. arXiv: `1605.06211`. URL: `http://arxiv.org/abs/1605.06211`.

[31] A. Srivastava, D. Jha, S. Chanda, U. Pal, H. D. Johansen, D. Johansen, M. A. Riegler, S. Ali, and P. Halvorsen. Msrf-net: a multi-scale residual fusion network for biomedical image segmentation, 2022. arXiv: `2105.07451 [eess.IV]`.

[32] G. Yang, H. Tang, M. Ding, N. Sebe, and E. Ricci. Transformer-based attention networks for continuous pixel-wise prediction, 2021. arXiv: `2103.12091 [cs.CV]`.

[33] X. Zhou, D. Wang, and P. Krähenbühl. Objects as points, 2019. arXiv: `1904.07850 [cs.CV]`.