Journal of
Imaging

Article

Topological Voting Method for Image Segmentation

Nga T. T. Nguyen [1,†] and Phuong B. Le [2,*,†]

[1] Torus Actions SAS, 31400 Toulouse, France; nttnga@torus-actions.fr
[2] Department of Mathematics, Hanoi University of Mining and Geology, Hanoi 100000, Vietnam
[*] Correspondence: lbphuong@sputnik.vn or lebichphuong@humg.edu.vn; Tel.: +84-988782112
[†] These authors contributed equally to this work.

Abstract: Image segmentation is one of the main problems in image processing. In order to improve the accuracy of segmentation, one often creates a number of masks (annotations) for a same image and then uses some voting methods on these masks to obtain a more accurate mask. In this paper, we propose a voting method whose voting rule is not pixel-wise but takes into account the natural geometric-topological properties of the masks. On three concrete examples, we show that our voting method outperforms the usual arithmetical voting method.

Keywords: voting method; image segmentation

check for updates

Citation: Nguyen, N.T.T.; Le, P.B. Topological Voting Method for Image Segmentation. *J. Imaging* **2022**, *8*, 16. https://doi.org/10.3390/jimaging8020016

Academic Editors: Xiaohao Cai, Ping Zhong and Gaohang Yu

Received: 13 November 2021
Accepted: 6 January 2022
Published: 18 January 2022

1. Introduction

Voting methods are ubiquitously used in human and artificial intelligence to improve the accuracy of automatic as well as hand-made annotations. The theoretical reason is simple: assuming that most individual annotators are relatively good (more often right than wrong) and independent to some extent, then for most situations, a majority of the annotators will be right while only a minority will be wrong, so by voting we are more likely to obtain a correct annotation in more situations than most individual annotators. Of course, when there are too many bad annotators who do not know what they are doing, then voting methods may be counter-productive: the results of good annotators will be killed by the results of bad annotators.

In this paper, we present our research on a novel voting method for the image segmentation problems in AI suggested to us by Professor Nguyen Tien Zung, Founder of Torus Actions SAS. This problem has been extensively explored in many papers [1–8] and the recently in the article [9]. The most popular voting method for image segmentation is the (soft or hard) arithmetical voting [4–6], where each pixel is voted by a majority rule for that pixel only and does not take into account the other pixels. Our starting idea is that the masks in each natural segmentation problem have a natural geometric-topological structure, the pixels are interrelated and not independent, so each pixel should be voted not individually but in connection with the other pixels. That is why we call our method the *topological voting method*. The idea of considering the structure of the mask rather than pixel-wise using in ensemble method has been studied in [1–3], where an image is divided into several regions (clusters) and the voting methods are applied for the regions. We shall discuss more about this idea in Section 2 of this paper.

We will present the topological voting algorithm, together with its variations, in Section 2. We will show not just one, but a whole family of topological voting methods, including *local* topological voting, and *hybrid* voting, which is a combination of arithmetical and topological voting methods, with a step to detect and exclude the outliers, those who are more likely to be wrong.

For simplicity, in this paper we will consider only the binary segmentation of 2D images, i.e., each image will be segmented into two parts: the region of interest, called the *mask*, and the rest (the background). To illustrate our ideas, we will work out three concrete examples: (1) segmentation of salt in seismic images; (2) segmentation of human faces in photos; and (3) segmentation of blood vessels in retinal images. These three examples are pretty typical of segmentation problems, and in all these examples, the experimental results, presented in Sections 3–5 of this paper, show that the topological voting method and its variations allow one to achieve better accuracy than the arithmetical voting method.

In Section 6, we offer some arguments, theoretical and experimental results that show *why* topological voting methods are efficient and give better results than arithmetical ones in many situations. These are the arguments and ideas that led to our experiments.

Section 7, the last section of this paper, is dedicated to conclusions and future work.

2. The Topological Voting Method

2.1. Image Segmentation and Jaccard Distance

Mathematically, one may represent a 2D-image binary segmentation tool or agent (a segmentor) as a discrete-valued map

$$S : \Omega \to \{0,1\}^{h\times w} \tag{1}$$

or a continuous-valued map

$$\tilde{S} : \Omega \to [0,1]^{h\times w} \tag{2}$$

from a space Ω of digital 2D images (of a fixed size, for simplicity), and $N = h \times w$ (height times width) is the number of pixels per image. For an image $x \in \Omega$, if $S(x)(i,j) = 1$, where (i,j) is the position of a pixel ($1 \le i \le h, 1 \le j \le w$), then it means that this pixel is in the mask of x made by the segmentor S; otherwise, the pixel is the background.

Convolutional neural networks (CNNs) that are used in image segmentation problems usually give us continuous-valued segmentors $\tilde{S}$, and we can obtain S from $\tilde{S}$ by fixing a threshold, for example

$$S(x)(i,j) = \begin{cases} 0 \text{ if } \tilde{S}(x)(i,j) < 0.5 \\ 1 \text{ if } \tilde{S}(x)(i,j) \ge 0.5 \end{cases} \tag{3}$$

We will assume that each image $x \in \Omega$ has a true mask (the ground truth) denoted by $S_{\text{true}}(x)$. We want to measure how good our segmentor is, i.e., how far is the mask $S(x)$ from the true mask $S_{\text{true}}(x)$.

A good and widely used measure of accuracy of binary segmentation is the so-called *intersection over union* (IOU) score, also known as the *Jaccard score*, introduced by Paul Jaccard in a paper in 1901 [10].

$$J(S(x), S_{\text{true}}(x)) = \frac{|S(x) \cap S_{\text{true}}(x)|}{|S(x) \cup S_{\text{true}}(x)|} \tag{4}$$

where $S(x) \cap S_{\text{true}}(x)$ denotes the intersection of the two masks $S(x)$ and $S_{\text{true}}(x)$ (i.e., the set of pixels where both segmentors have value equal to 1), $S(x) \cup S_{\text{true}}(x)$ denotes their union (where at least one of them has value equal to 1), and the absolute value sign denotes the surface area (i.e., the number of pixels in the set).

Remark that an equivalent way to write the Jaccard score is MOM (min over max):

$$J(S(x), S_{\text{true}}(x)) = \frac{\sum_{i,j} \min(S(x)(i,j), S_{\text{true}}(x)(i,j))}{\sum_{i,j} \max(S(x)(i,j), S_{\text{true}}(x)(i,j))} \tag{5}$$

An advantage of Formula (5) over Formula (4) is that it also works for *soft masks*, i.e., for continuous-valued segmentors: if U and V are two soft masks then we can define their relative *soft Jaccard score* as

$$J(U,V) = \frac{\sum_{i,j} \min(U(i,j), V(i,j))}{\sum_{i,j} \max(U(i,j), V(i,j))} \tag{6}$$

The *Jaccard distance* of a mask $S(x)$ to a true mask $S_{true}(x)$ is defined by the formula

$$d_{Jaccard}(S(x), S_{\text{true}}(x)) = 1 - J(S(x), S_{\text{true}}(x)), \tag{7}$$

and it measures "how far" $S(x)$ is from $S_{\text{true}}(x)$. The two masks coincide if, and only if, the Jaccard distance between them is equal to 0. In the case of soft masks, the same formula still works and gives what we call the *soft Jaccard distance*.

In this paper, we will mainly use the Jaccard score (and the Jaccard distance) to measure the accuracy of our automatic segmentors.

Let us mention that there is a naive *binary accuracy score*, which is by definition the number of pixels at which $S(x)$ coincides with $S_{\text{true}}(x)$ divided by the total number of pixels N. When the true mask is small (the problem of segmentation of small objects), it may happen that the segmentor S gives a mask which is completely different from the true mask (no intersection) and the binary accuracy score is still near 1 (maximal possible), so this binary accuracy score is not a very good measure of accuracy, though we may still compute it sometimes.

2.2. Arithmetical Voting

In automatic as well as hand-made segmentation, one often creates not just one, but many segmentors $S_1, \ldots, S_n$ for the same problem, using different CNNs, or different training datasets, or different data augmentation methods, etc., and then ensembles them by a voting method to hopefully obtain a segmentor which is more accurate on the average than each one of them. The most obvious voting method is the majority vote: for each pixel, each segmentor has one vote, and the candidate value (0 or 1) that has the most votes wins. This majority voting method is also called the *hard arithmetical voting*, and there is another variant of arithmetical voting called the *soft voting*, see, e.g., [4,5]. In soft voting, one uses continuously-valued segmentors $\tilde{S}_1, \ldots, \tilde{S}_n$ instead of discrete-valued segmentors $S_1, \ldots, S_n$, and put

$$\tilde{S}_{\text{voted}} = \frac{1}{n} \sum_{k=1}^{n} \tilde{S}_k \tag{8}$$

when n is large, then by the law of large numbers, soft voting and hard voting will give more or less the same results. However, when n is small, soft voting may be finer and give slightly better results than hard voting. One may fine-tune the above arithmetical voting formula by giving different weights to different segmentors (weighted averaging formula).

Our *topological voting methods* are very different from arithmetical voting. In the following Sections 2.3–2.5, we present three versions of the method. One of them—the simplest version of the proposed method, turns out to be the same as the "Best of K" method in [2], though we arrived at it independently. The other two versions are different.

2.3. Topological Voting: Simplest Version

The simplest forms (hard and soft topological voting) are presented in the following. The hard version is the one called "Best of K" in [2] with only one cluster that is the whole image. More details, they both consist of the following steps:

(i) For an input image x, take n masks $S_1(x), \ldots, S_n(x)$ given by n different segmentors $S_1, \ldots, S_n$;

(ii) For each index $k \in \{1, \ldots, n\}$, measure the total distance from $S_k(x)$ to the other masks, with respect to some natural distance function.

There are different natural distances in geometry that fit different problems. For example, the Hausdorff distance, also known as Hausdorff–Pompeiu distance, can be used effectively in many image processing problems [11,12]. For simplicity, here, we will only use the (soft or hard) Jaccard distance in our experiments and define the total distance $d_k(x)$ from $S_k(x)$ to the other masks by the following formula:

$$d_k(x) = \sum_{i=1}^{n} d_{Jaccard}(S_i(x), S_k(x)) \tag{9}$$

(iii) (Winner takes all) The mask with the smallest total Jaccard distance to the other masks wins, i.e.,

$$S_{\text{voted}}(x) = S_l(x) \text{ where } l = \arg\min_k d_k(x) \tag{10}$$

An equivalent way to formulate the "winner takes all" rule is: the mask with the highest total Jaccard score wins, i.e.,

$$S_{\text{voted}}(x) = S_l(x) \text{ where } l = \arg\max_k J_k(x) \tag{11}$$

and

$$J_k(x) = \sum_{i=1}^{n} J(S_i(x), S_k(x)) \tag{12}$$

In *soft topological voting*, one uses the same formulas as above but applied to the soft masks instead of the hard masks. A technical side notes: it may be useful to regularize the sigmoid values of the soft masks before computing soft scores, for example, by truncating them at 0.2 and 0.8: $s_{regularized} = 0$ if $0 \leq s \leq 0.2$, $s_{regularized} = 10(s - 0.2)/6$ if $0.2 \leq s \leq 0.8$, and $s_{regularized} = 1$ if $0.8 \leq s \leq 1$. This is actually what we do with soft masks in our experiments.

It turns out that, in many cases, the above "winner takes all" simple topological voting method already gives results that are superior to the arithmetical voting method. We will show it in the example of salt segmentation in seismic images (see Section 3).

The following is a diagram for the system:

Topological Voting Schema (Simplest Version)

Candidates $Y_1, Y_2, Y_3, ..., Y_n$ (Topological objects)

Topological distances $d_{ij} = d(Y_i, Y_j)$

Total distance from Y_i to the rest

$$d_i = \Sigma_j d_{ij}$$

Selected candidate Y *selected*, where *selected* $= \arg\min_i d_i$

2.4. Local Topological Voting

Some objects (e.g., blood vessels) have a complicated global structure and so even a good segmentation may be imprecise in many places. To improve the accuracy of annotation, one should in theory reduce the complexity of the things to annotate by decomposing them into smaller, simpler things. For image segmentation, it means that we can often cut a big complicated image into smaller, simpler ones, easier to segment. This 'localized segmentation' strategy together with the topological voting idea leads to what we call *local topological voting*.

To be more concrete, the local topological voting algorithm that we will use for our experiments presented in this paper is the following:

(i) Fix a natural number s, which will be the *radius* of the local neighborhoods;

(ii) For each pixel, consider the neighborhood of radius s around that pixel: If the pixel is at position (x, y), then its s-neighborhood will be the square $[x - s, x + s] \times [y - s, y + s]$. (If the pixel is near the border, then its s-neighborhood will be the intersection of this square with the image);

(iii) Use the topological voting algorithm, as presented in the previous subsection, on the predicted masks restricted to the s-neighborhood of the pixel (x, y) to obtain the result for this pixel. In other words, the voted value of the pixel (x, y) is equal to the value at (x, y) of the annotator which is considered to be locally topologically the best in the s-neighborhood of (x, y);

(iv) Do the above step (iii) for every pixel to obtain the total mask.

The above local topological algorithm contains a parameter s, which is the size of the neighborhoods, i.e., the degree of locality. Notice that, when $s = 0$, then the s-neighborhood of a pixel is just itself, no topological structure is taken into account, and the 0-neighborhood topological voting is just the usual arithmetical voting. At the other extreme, when s is greater than the size of the image, then any s-neighborhood is the whole image, and we return back to the first version of our topological voting method. By varying s from 0 to infinity, we obtain a whole family of voting methods that goes from arithmetical (pixel-by-pixel) to topological (whole-picture) voting.

Intuitively, for each problem there is an optimal neighborhood size, and the two extreme sizes (0 and infinity) are not the best ones. In our experiments, we will vary the radius s, and not surprisingly, we will see that some neighborhood sizes are really better than the others (see Section 5 on blood vessel segmentation).

We illustrate an example of this voting version in Figure 1. In that example, the radius is fixed $s = 1$. At each pixel, a window of that radius is applied for all the segmentators $S1, S_2, ..., S_n$ to obtain n windows. We then use the method described in Section 2.3 for the n windows to obtain the final window. The value of the center pixel of the final window is chosen as the result for the global mask. We repeat these steps through all pixels of the image to obtain the final mask.

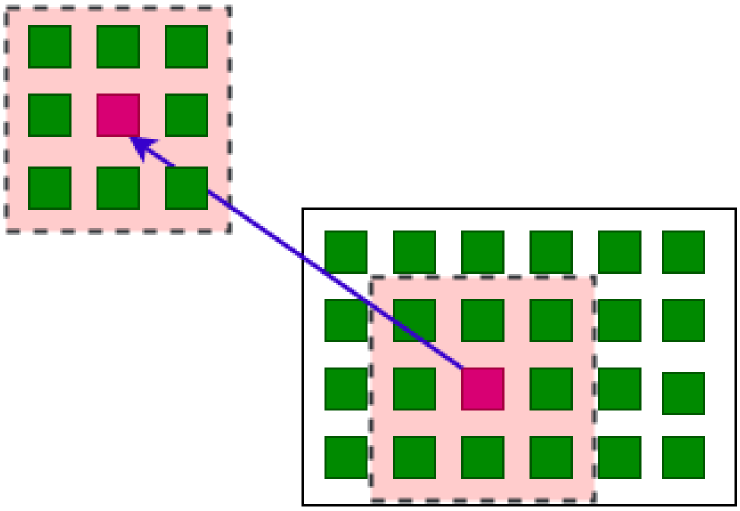


Figure 1. An example for local topological voting $s = 1$.

2.5. Hybrid Topological-Arithmetical Voting

The hybrid voting that we present in this paper is a 2-round voting. An example is illustrated in Figure 2 and it goes as follows:

(i) In the first round, we use the (local or global) topological voting, not to choose the winner, but to choose the losers to exclude from the race.

More concretely, in this paper we will use one of the two following simple exclusion rules for each hybrid voting:

- Method 1: Choose a threshold $H > 1$. For each input x keep the masks $S_k(x)$ such that $J_k(x) \geq n/H$, where $J_k(x) = \sum_{i=1}^{n} J(S_k(x), S_i(x))$ and n is the total number of our individual segmentors. If $J_k(x) < n/H$ then $S_k(x)$ considered to be an "outlier" and is excluded from the second round;

- Method 2: Choose a number $1 \leq n_{\text{select}} \leq n$. For each input x keep the n_{select} masks $S_k(x)$ with the highest Jaccard scores $J_k(x)$, and exclude the other $n - n_{\text{select}}$ masks.

In method 1, the number of masks admitted to the second round is not fixed and it is different for each input, while in method 2, this number is always equal to n_{select}.

(ii) The candidates that remain in the second round will be voted arithmetically.

Another variation of hybrid voting is voting with weights determined by the distance function, or equivalently, by the Jaccard scores $J_1(x), \ldots, J_n(x)$: the higher the Jaccard score $J_k(x)$ is (relatively to the Jaccard scores of the other annotators), the higher the weight of $S_k(x)$ will be in the weighted arithmetical voting formula. Topological voting is when all the weights are equal to 0 except one weight which is equal to 1.

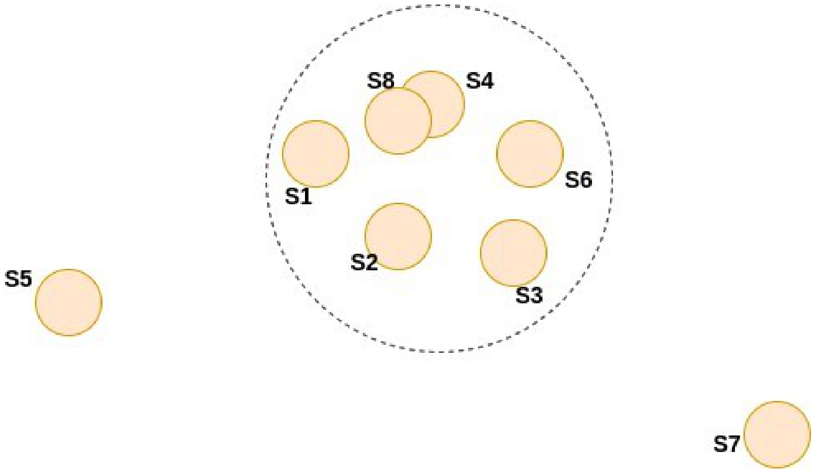


Figure 2. An example for hybrid voting. At round-1, segmentator 5 and segmentator 7 will be excluded since they are far from the others, the rest are kept for round-2 using (either soft or hard) arithmetical voting.

3. Salt Segmentation in Seismic Images

The automatic segmentation of salt deposits in seismic images is an important problem for geology companies in search of hydrocarbons. In 2018, Kaggle organized the "TGS Salt Identification Challenge" on this problem, and provided a training dataset of 4000 annotated grayscale images of size 101 × 101 [13].

Our aim here is not to build a "state-of-the-art" automatic segmentor for the Kaggle challenge (in real life, one will not cut the seismic images into very small pieces and compute the accuracy scores the way Kaggle did anyway), but just to do experiments on the topological voting method. For that purpose, we will build our AI models based on a popular light-weight convolutional neural network architecture called MobileNet [14], which is very handy in the sense that one can train it very fast, and it can run on small devices, such as mobile phones. We train our models using Tensorflow and Keras [15]. The loss function that we use in training our model is the sum of the binary cross entropy and the Dice loss function [16]. We use random padding, translation, rotating, flipping, and cutting, to create square images of size 128 × 128 from the original images of size 101 × 101, since our MobileNet models use inputs of size 128 × 128.

We divide 4000 annotated images into 2 sets: the training set (3000 images) and the test set (1000 images). The results that we show in the tables below are for the test set (not used in the training process of course). The training set is then divided into 5 folds, each fold containing 600 images. For each fold, we train a corresponding AI model, which uses that fold for validation and the other four folds for training. We train each model 500 epochs, each epoch has 3000 inputs, so in total each model is trained on 1,500,000 inputs. Each input is an image taken randomly from the training set and then undergoing random transformations (augmentations in data pre-processing). We do not take the AI model after exact 500 epochs of training, but rather the AI model after the epoch that offers the highest Jaccard scores on validation among all the 500 epochs.

After the above training process, we obtain 5 AI segmentors, corresponding to our 5 folds. Then, for each image in the test set, we vote on the 5 masks given by these 5 segmentors, using arithmetical voting and the (soft and hard) topological voting methods. The results are shown in Table 1. Here, the Jaccard score is the means Jaccard score over the test set.

Table 1. Comparison of voting methods on 5 folds for the Salt dataset.

AI Model	Hard Jaccard Score	Interval for p (N = 1000, c = 90%)
Fold 1	0.7253	(0.7014, 0.7480)
Fold 2	0.7360	(0.7124, 0.7583)
Fold 3	0.7240	(0.7001, 0.7467)
Fold 4	0.7404	(0.7169, 0.7626)
Fold 5	0.7429	(0.7195, 0.7650)
Soft Arithmetical Voting	0.7813	(0.7590, 0.8020)
Hard Topological Voting	0.7831	(0.7608, 0.8038)
Soft Topological Voting	0.7847	(0.7625, 0.8053)

Some concrete examples of the masks given by our five individual segmentors, and the results of two different voting methods (arithmetical and topological) and shown in Figures 3–6, together with the original images and the true masks (the ground truth given in the dataset). These four figures illustrate how the topological voting method works differently from the arithmetical one. We can see in these figures that the arithmetical chooses the one that equals to average of all mask pixel-wise regardless the structure of each individual mask, consequently, it causes a final voting result which is unstructured regarding that the region of salt is often smooth and continuous. Meanwhile, the topological chooses the mask which is the most common to the others in the meaning that it has the smallest "distance" to the other masks and it is thus reasonable close to the true mask.

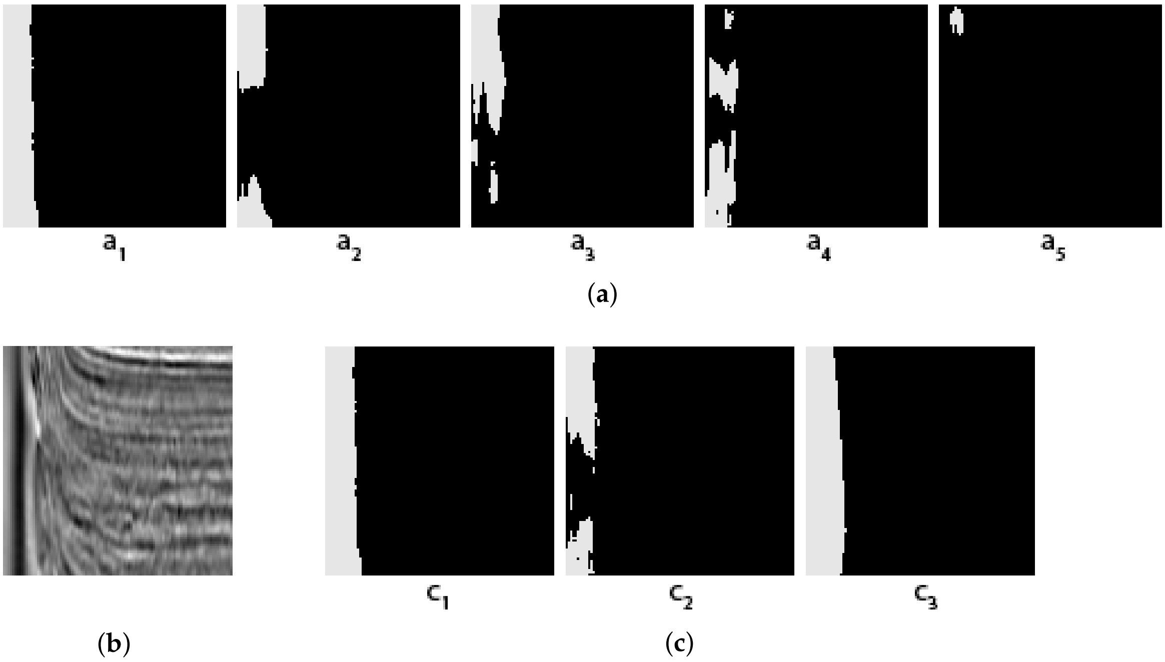


Figure 3. Topological vs. Arithmetical voting in salt segmentation—The first example. (**a**) (Masks created by 5 individual segmentors (from 5 folds); (**b**) Original image; (**c**) C_1: Topological voting, C_2: Arithmetical voting, C_3: True mask.

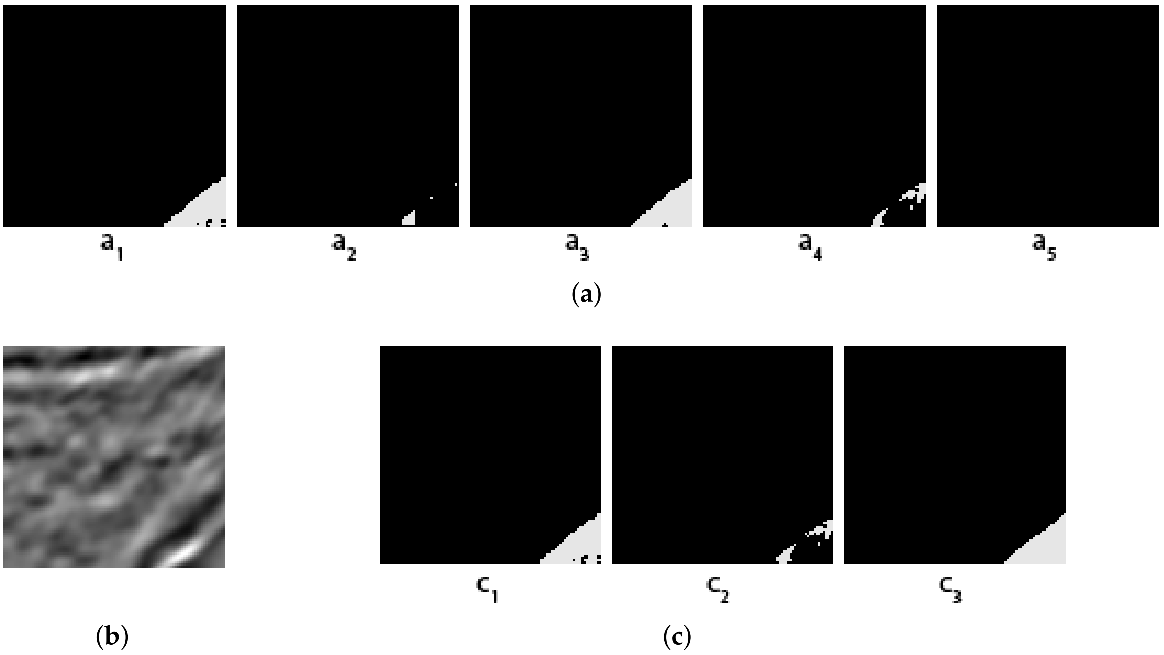


Figure 4. Topological vs. Arithmetical voting in salt segmentation -The second Example. (**a**) Masks created by 5 individual segmentors (from 5 folds); (**b**) Original image; (**c**) C_1: Topological voting, C_2: Arithmetical voting, C_3: True mask.

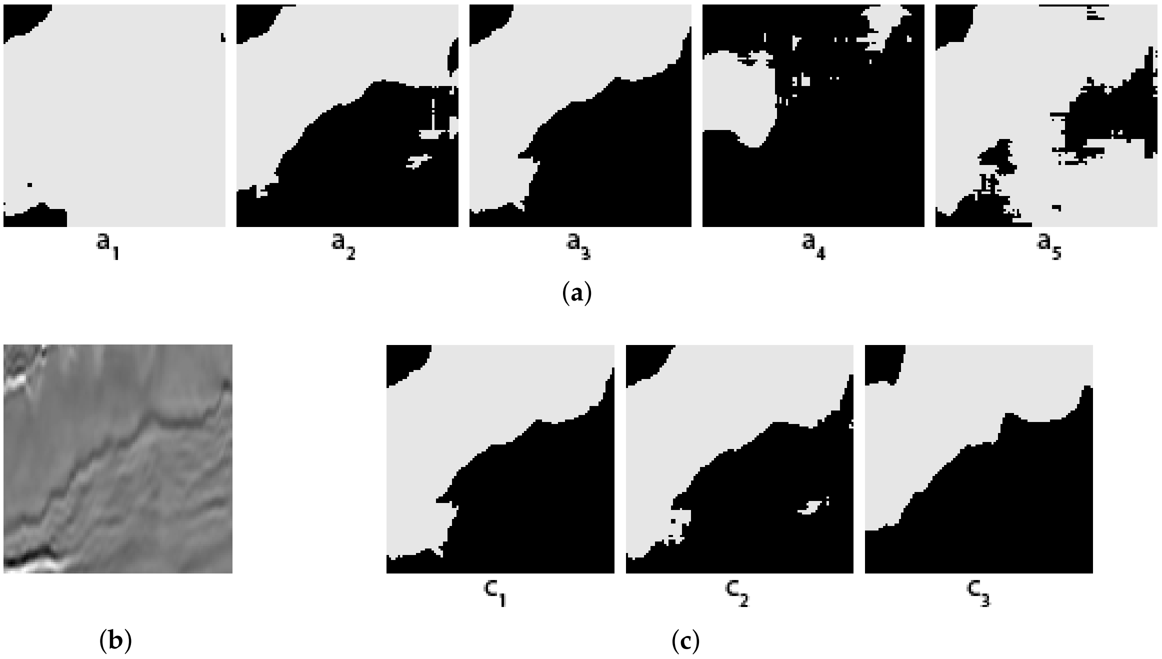


Figure 5. Topological vs. Arithmetical voting in salt segmentation - The third example. (**a**) Masks created by 5 individual segmentors (predictions of the 5 folds); (**b**) Original image; (**c**) C_1: Topological voting, C_2: Arithmetical voting, C_3: True mask.

In order to improve the accuracy, one can increase the number of individual segmentors. So we created 5 additional AI models using 5 additional folds, in the same way as before. The ensemble results using 10 models, shown in Table 2, are indeed better than the ensemble results using just 5 models.

Table 2 also shows that, for the salt segmentation problem, the topological voting method clearly beats the arithmetical voting method (by more than a full percentage point).

This table also shows the results of the hybrid topological-arithmetical voting method (at different thresholds), which are slightly better than the simple topological voting method.

Table 2. Comparison of voting methods on 10 folds for the Salt dataset.

AI Model	**Hard Jaccard Score**	**Interval for p (N = 1000, c = 90%)**
Fold 1	0.7253	(0.7014, 0.7480)
Fold 2	0.7360	(0.7124, 0.7583)
Fold 3	0.7240	(0.7001, 0.7467)
Fold 4	0.7404	(0.7169, 0.7626)
Fold 5	0.7429	(0.7195, 0.7650)
Fold 6	0.7371	(0.7135, 0.7594)
Fold 7	0.7479	(0.7246, 0.7699)
Fold 8	0.7358	(0.7122, 0.7581)
Fold 9	0.7426	(0.7192, 0.7647)
Fold 10	0.7086	(0.6843, 0.7317)
Arithmetical (soft) voting	0.7877	(0.7656, 0.8082)
Topological (soft) voting	0.7996	(0.7779, 0.8197)
Topological (hard) voting	0.8001	(0.7784, 0.8201)
Hybrid voting (threshold = 1.2)	0.8009	(0.7793, 0.8209)
Hybrid voting (threshold = 2)	0.8018	(0.7802, 0.8218)
Hybrid voting (threshold = 3)	0.8027	(0.7811, 0.8226)
Hybrid voting (threshold = 4)	0.7998	(0.7781, 0.8199)
Hybrid voting ($n_{\text{selected}} = 4$)	0.8019	(0.7803, 0.8219)
Hybrid voting ($n_{\text{selected}} = 5$)	0.8000	(0.7783, 0.8200)
Hybrid voting ($n_{\text{selected}} = 7$)	0.7979	(0.7762, 0.8180)

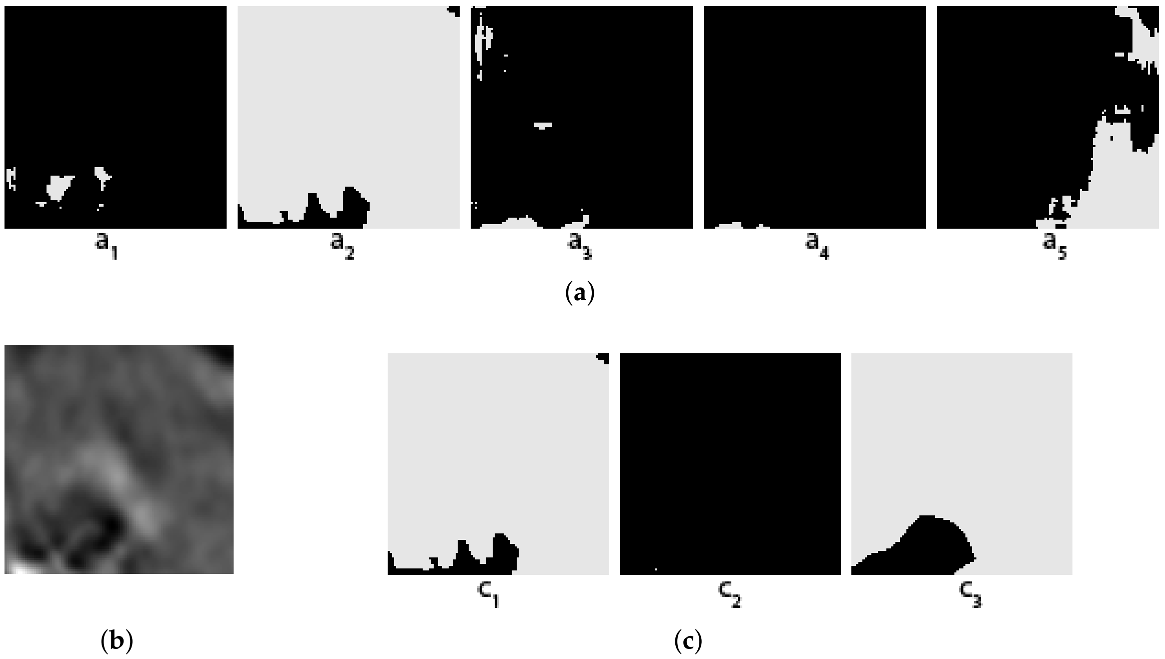


Figure 6. Topological vs. Arithmetical voting in salt segmentation—The fourth example. (**a**) Masks created by 5 individual segmentors (from 5 folds); (**b**) Original image; (**c**) C_1: Topological voting, C_2: Arithmetical voting, C_3: True mask.

As a side remark, we note that, if we measure the performance by using the binary accuracy metric instead of the Jaccard score, then the scores will be very high even for completely wrong segmentations, and the topological voting method will give worse results than the arithmetical voting method if we use this binary accuracy metric, see Table 3.

Table 3. Comparison of voting methods on 10 folds for the Salt dataset, using binary accuracy score.

AI Model	Binary Accuracy Score	Interval for p (N = 1000, c = 90%)
Fold 1	0.9334	(0.9192, 0.9453)
Fold 2	0.9356	(0.9215, 0.9473)
Fold 3	0.9373	(0.9234, 0.9488)
Fold 4	0.9288	(0.9142, 0.9411)
Fold 5	0.9315	(0.9171, 0.9435)
Fold 6	0.9297	(0.9152, 0.9419)
Fold 7	0.9374	(0.9235, 0.9489)
Fold 8	0.9296	(0.9151, 0.9418)
Fold 9	0.9306	(0.9161, 0.9427)
Fold 10	0.9367	(0.9228, 0.9483)
Arithmetical voting	0.9421	(0.9287, 0.9531)
Topological voting (using binary accuracy)	0.9412	(0.9277, 0.9523)
Hybrid voting (threshold = 1.5)	0.9417	(0.9282, 0.9527)
Hybrid voting (threshold = 2)	0.9418	(0.9283, 0.9528)
Hybrid voting (threshold = 3)	0.9421	(0.9287, 0.9531)

4. Human Face Segmentation in Photos

The Face and Skin Detection Database which contains 4000 images is created by S. L. Phung, T. Y. Ke, and F. H. C. Tivive and used in [17] to support research on skin segmentation and face recognition. The dataset provides many types of ground-truth, such as human face segmentation and skin segmentation. Here, we will use the human face segmentation of this dataset.

Actually, the masks that we will use are not exactly the human faces, but the smallest rectangles containing them. To make the problem more interesting, we will partially hide the human faces on the photos by random rectangles, and let the machine learn to segment the full human faces despite those hidden parts. Figure 7e shows the original image in which the faces are covered by a random rectangle and Figure 7f (the rightmost image) shows the mask of that image.

The dataset of 4000 photos is divided into 2 subsets: 1000 images for testing, and 3000 images for training. The 3000 training images are divided into 10 folds, each fold contains 600 images, so the folds overlap: the first 5 folds are a partition of the training data, and the second 5 folds are another partition as well.

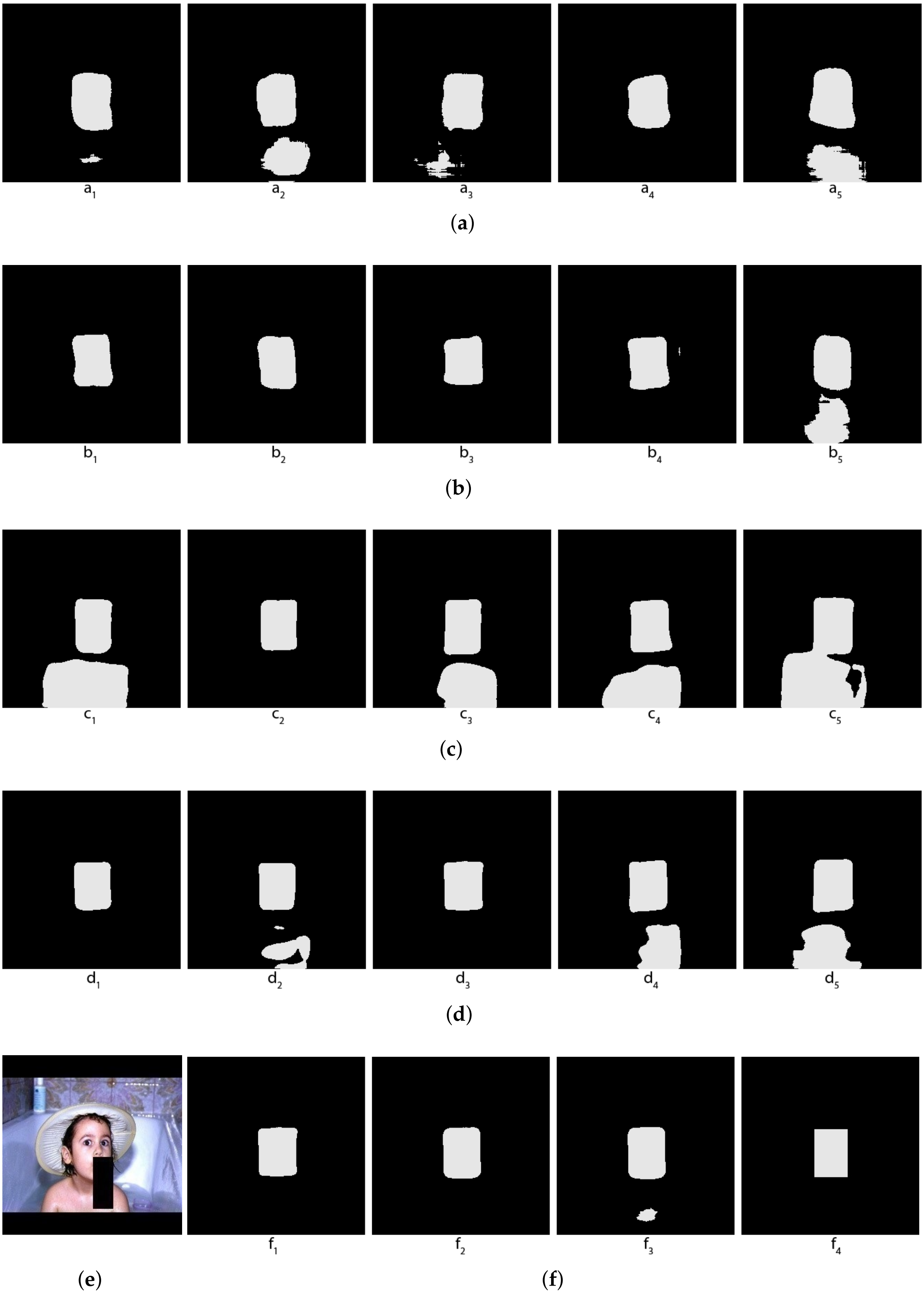
The original photos in the dataset are of different sizes, but we will augment and resize them into images of size 256 × 256 before feeding them into our CNN models. The transformations that we use to augment the images are the standard ones: random rotation, flipping, brightness modification, cropping, resizing, padding, and noise adding. As mentioned above, we also add a random back rectangle to each image to partially cover the faces (without changing the masks).

We use two different CNN architectures for our experiment: the light-weight MobileNet [14], and the more heavy-weight EfficientNet B4 [18], so in total, we obtain 20 different AI segmentors for our human face segmentation problem.

It is interesting to look at various inputs and outputs to see how the individual segmentors perform, and what are their main structural mistakes. For example:

In Figure 7, one can see that some individual segmentors mistake body skin for facial skin, while the other segmentors do not make this mistake. Topological and hybrid voting allows us to exclude those segmentors who make this mistake, and so the voted result does not contain body skin in the mask, while the arithmetical voting does not have this semantic advantage, and the result of the arithmetical voting still contains body skin.

In Figure 8, one can see that some individual segmentors barely recognize any facial skin, while there is one segmentor that mistakes a shirt for facial skin. Figure 9 is another very interesting example, where some segmentors mistake a dog face for a human face.



(**a**)

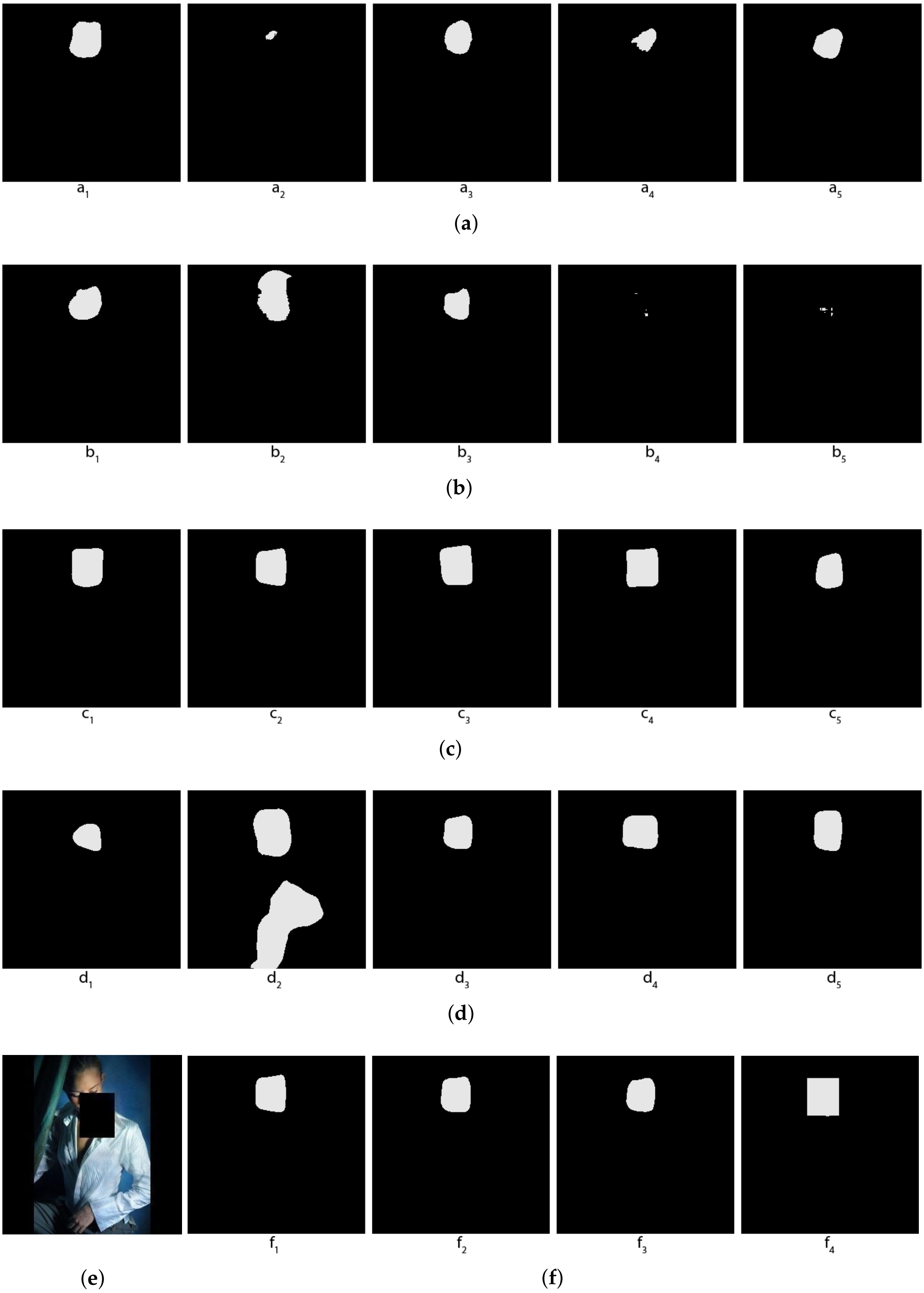
(**b**)

(**c**)

(**d**)

(**e**) (**f**)

Figure 7. Comparison of voting methods for human faces—The first example. (**a**) Predictions of the first 5 folds (with Mobilenet); (**b**) Predictions of the last 5 folds (with Mobilenet); (**c**) Predictions of the first 5 folds (with EfficienNet); (**d**) Predictions of the last 5 folds (with EfficienNet); (**e**) Original image; (**f**) f_1: Topological voting, f_2: Hybrid voting (threshold = 2), f_3: Arithmetical voting, f_4: True Mask.



(**a**)

(**b**)

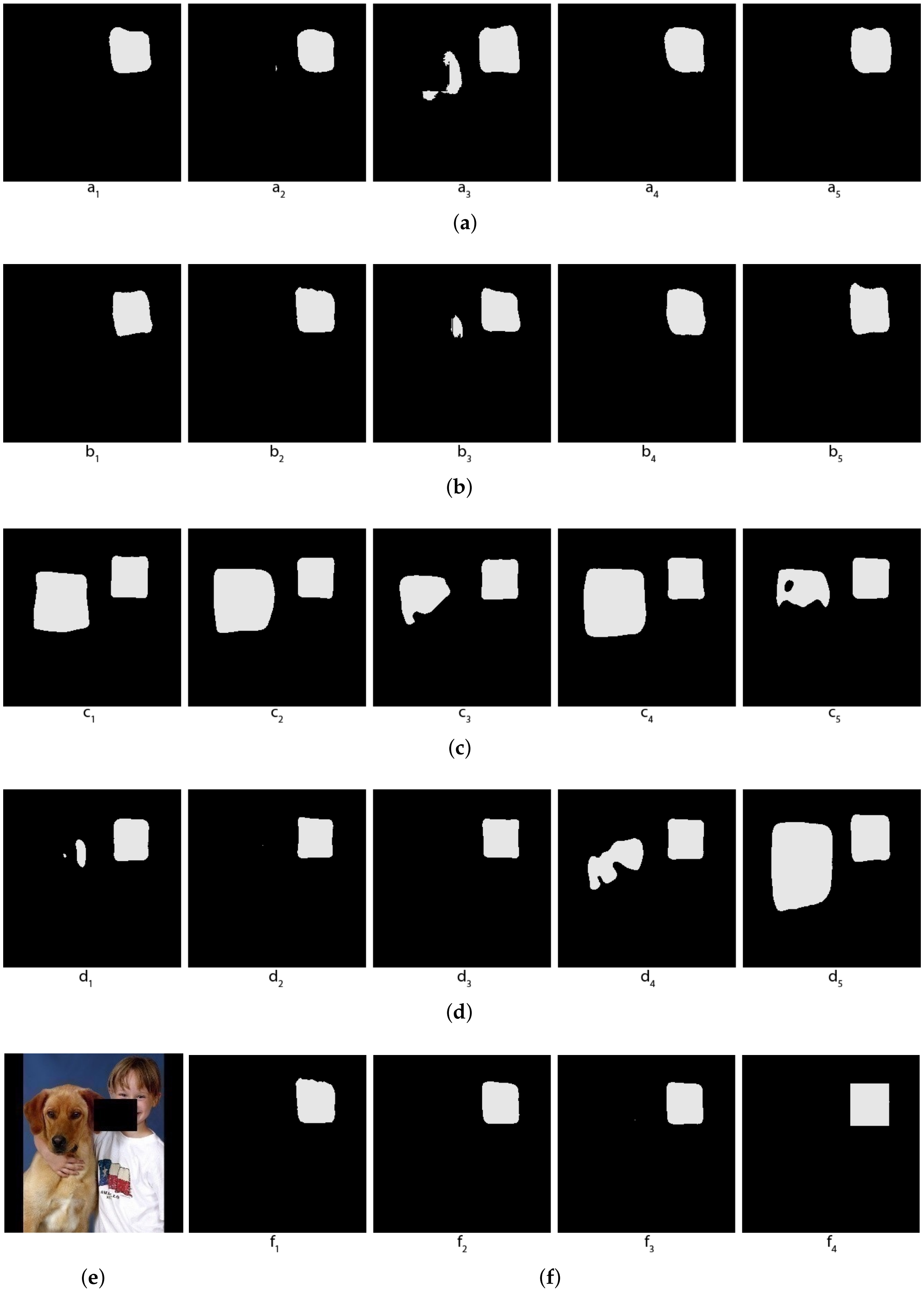
(**c**)

(**d**)

(**e**)

(**f**)

Figure 8. Comparison of voting methods for human faces—The second example. (**a**) Predictions of the first 5 folds (with Mobilenet); (**b**) Predictions of the last 5 folds (with Mobilenet); (**c**) Predictions of the first 5 folds (with EfficienNet); (**d**) Predictions of the last 5 folds (with EfficienNet); (**e**) Original image; (**f**) f_1: Topological voting, f_2: Hybrid voting (threshold = 2), f_3: Arithmetical voting, f_4: True Mask.



(**a**)

(**b**)

(**c**)

(**d**)

(**e**) (**f**)

Figure 9. Comparison of voting methods for human faces—The third example. (**a**) Predictions of the first 5 folds (with Mobilenet); (**b**) Predictions of the last 5 folds (with Mobilenet); (**c**) Predictions of the first 5 folds (with EfficienNet); (**d**) Predictions of the last 5 folds (with EfficienNet); (**e**) Original image; (**f**) f_1: Topological voting, f_2: Hybrid voting (threshold = 2), f_3: Arithmetical voting, f_4: True Mask.

The accuracy scores of different voting methods on our 20 models are given in Table 4. It shows that hybrid voting gives the best results, by excluding those segmentors that make very gross mistakes.

Table 4. Comparison of voting methods on 10 folds for the Face dataset.

	MobileNet Model	**EfficientNet Model**
Fold 1	0.7572	0.7827
Fold 2	0.7495	0.7845
Fold 3	0.7525	0.7843
Fold 4	0.7527	0.7805
Fold 5	0.7625	0.7849
Fold 6	0.7588	0.7878
Fold 7	0.7611	0.7879
Fold 8	0.7553	0.7917
Fold 9	0.7637	0.7788
Fold 10	0.7604	0.7795
Arithmetical (soft) Voting	0.7909	0.8035
Topological (soft) Voting	0.7832	0.7990
Topological (hard) Voting	0.7833	0.7985
	Score of voting on all 20 segmentors	
		Interval for p (N = 1000, c = 90%)
Arithmetical (soft) Voting	0.8088	(0.7875, 0.8285)
Topological (soft) Voting	0.8043	(0.7828, 0.8242)
Topological (hard) Voting	0.8032	(0.7816, 0.8231)
Hybrid voting (threshold = 1.5)	0.8092	(0.7879, 0.8289)
Hybrid voting (threshold = 2)	0.8093	(0.7880, 0.8289)
Hybrid voting (threshold = 3)	0.8092	(0.7879, 0.8289)
Hybrid voting ($n_{\text{select}} = 10$)	0.8085	(0.7871,0.8282)
Hybrid voting ($n_{\text{select}} = 15$)	0.8090	(0.7877, 0.8287)
Hybrid voting ($n_{\text{select}} = 17$)	0.8091	(0.7878, 0.8287)

One may wonder why the plain-vanilla topological voting method gives lower scores than the arithmetical voting method for the above experiment of facial segmentation? We think that the reason lies in the fact that the masks themselves are very simple (just rectangles), and the topological voting method excels at more complicated masks.

5. Blood Vessel Segmentation

The problem of segmentation of tree-like structures, such as microglia extensions, neurovascular structures, blood vessels, and pulmonary trees, are of great interest in medical AI, see, e.g., [19] and references therein. Rouchdy and Cohen [19] studied the problem of segmentation of blood vessels in retinal images using a method called *geodesic voting with radius* (no deep learning), and showed the superiority of their method to older approaches such as the edge-based level set method [20], the Chan and Vese method [21], and the fuzzy connectedness method [22]. The database that they used is the digital retinal images for vessel extraction (DRIVE) data [23].

In this section, we propose to use deep learning to solve the retinal blood vessel segmentation problem using this same DRIVE dataset. Again, we will use the light-weight MobileNet [14]. Not surprisingly, the deep neural networks can give better segmentation results than the previous image processing methods, including the geodesic voting method [19].

The DRIVE dataset contains 20 images of size 565 × 584. We first divide it into two sets: 15 images for training and 5 images for testing. We create 15 folds for the training and the cross-validation of our AI using the 15 training images. Each fold leads to one segmentor, so in total we have 15 individual segmentors to vote on. Each fold uses 13 images for training and 2 images for validation. Each image is then transformed (augmented) using a combination of operations, including random rotating, flipping, padding, brightness modification, noise adding, cropping, and padding, into square images of size 256 × 256, before being fed into our CNN based on MobileNet for training and validation. The five original images used for testing are cropped into 2000 images of size $x \times y$ where x and y are random whole numbers in the interval $[220, 256]$ and then padded to square images of size 256 × 256, so our test set consists of 2000 images of size 256 × 256.

Figure 10 shows an example of our 15 individual segmentations of an image from the test set, and the results of three different voting methods on these 15 segmentations compared to the true mask. One can see visually on these pictures that both the local topological method and the hybrid voting methods give better results than the arithmetical voting methods: fewer missing pixels compared to the true mask (the vessels are less broken).

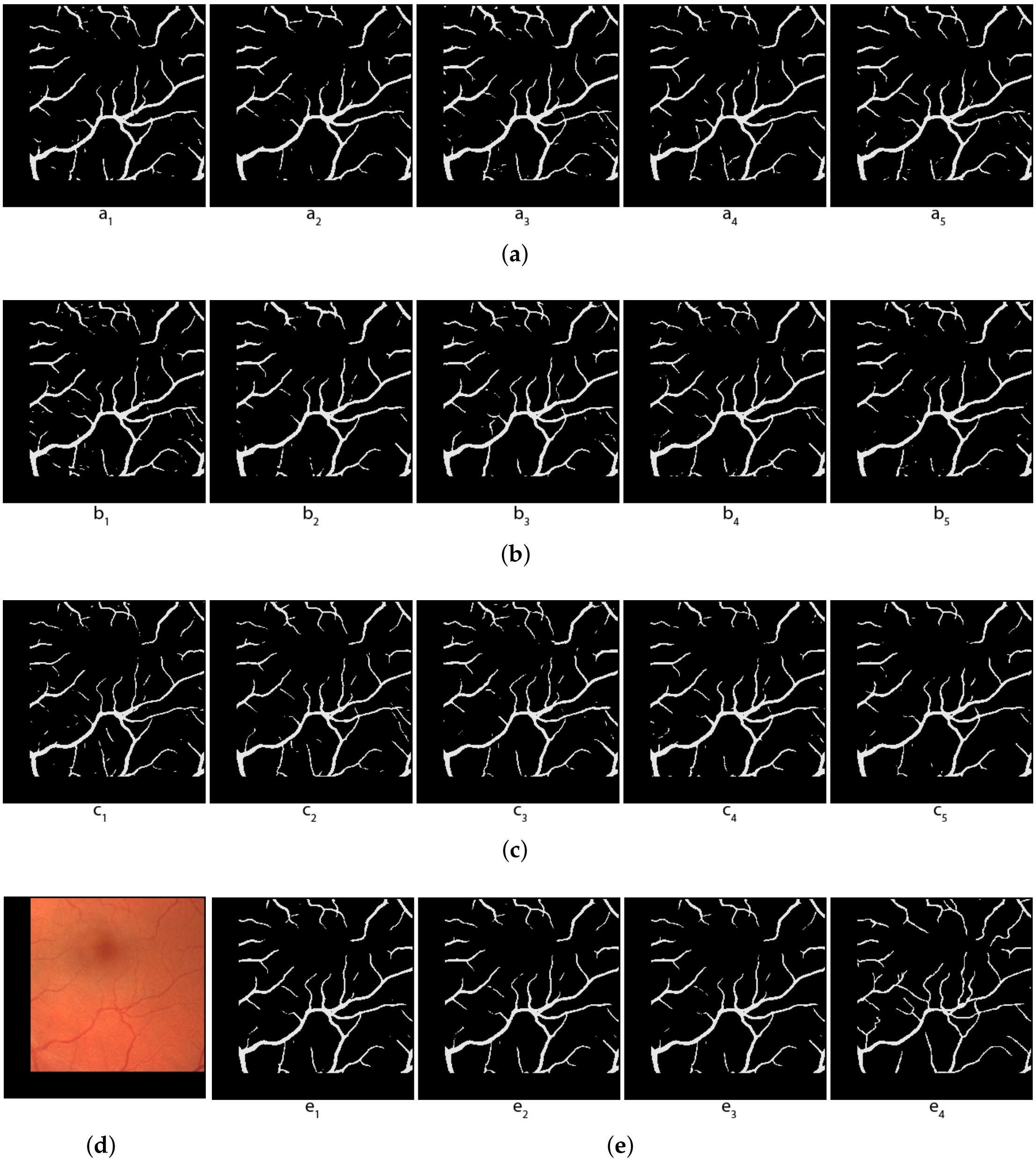


Figure 10. Comparison of voting methods for blood vessel segmentation. (**a**) Predictions of the first 5 folds; (**b**) Predictions of the second 5 folds; (**c**) Predictions of the last 5 folds; (**d**) Original image; (**e**) e_1: Local topological voting with $s = 10$, e_2: Hybrid voting ($s = 10$, threshold = 2), e_3: Arithmetical voting, e_4: True Mask.

Table 5 shows a comparison of voting methods on the DRIVE dataset.

Table 5. Comparison of voting methods on 15 folds for the DRIVE dataset.

AI Model	Hard Jaccard Score	Interval for *p* (N = 2000, c = 90%)
Fold 1	0.6159	(0.5978, 0.6337)
Fold 2	0.6339	(0.6160, 0.6515)
Fold 3	0.6116	(0.5935, 0.6294)
Fold 4	0.6197	(0.6016, 0.6374)
Fold 5	0.6282	(0.6102, 0.6458)
Fold 6	0.6224	(0.6043, 0.6401)
Fold 7	0.6105	(0.5924, 0.6283)
Fold 8	0.6105	(0.5924, 0.6283)
Fold 9	0.6175	(0.5994, 0.6353)
Fold 10	0.6145	(0.5964, 0.6323)
Fold 11	0.6204	(0.6023, 0.6381)
Fold 12	0.6133	(0.5952, 0.6311)
Fold 13	0.6139	(0.5958, 0.6317)
Fold 14	0.6241	(0.6061, 0.6418)
Fold 15	0.6117	(0.5936, 0.6295)
Local topological voting (radius = 20)	0.6347	(0.6168, 0.6523)
Arithmetical (soft) voting	0.6364	(0.6185, 0.6540)
Local topological voting (radius = 10)	0.6384	(0.6205, 0.6560)
Local topological voting (radius = 5)	0.6391	(0.6212, 0.6566)
Hybrid voting (radius = 20, threshold = 2)	0.6399	(0.6220, 0.6574)
Hybrid voting (radius = 5, threshold = 2)	0.6412	(0.6233, 0.6587)
Hybrid voting (radius = 10, threshold = 2)	0.6418	(0.6239, 0.6593)

One may notice that the Jaccard scores obtained in blood vessels segmentation are much lower than in the previous two problems. The main reason is simple: the vessels are very thin, and so the Jaccard scores are very sensitive to small variations in the segmentation. Another reason is that MobileNet is a light-weight CNN aimed at quick processing and not highest accuracy, and we did not do any special optimization here either. What is important for us here is the fact that the gain obtained by the hybrid topological-arithmetical voting method is significant compared to the arithmetical voting method.

6. Why Does Topological Voting Work?

Let us recall that the philosophical reason behind our topological and hybrid voting methods is the following: for meaningful objects, their masks must have certain geometrical or topological properties or shapes. Assuming that we have good (but not yet excellent) segmentors $S_1, \ldots, S_n$, most masks given by them (for a typical input) will have reasonably good shapes reflecting topological-geometrical properties of the segmented object and are sufficiently close to the true mask. Consequently, in general, a mask which is far from the other masks will also be far from the true mask, while a mask which is closer to most other masks will also be closer to the true mask. So, even though we do not know what the true mask is, we can use the total distance function as a proxy to estimate how far is $S_k(x)$ from the true mask.

As a side remark, one may see a parallel between the different voting methods used in image segmentation and the voting methods used in politics. For example, elections of village mayors are similar to pixel-by-pixel voting (each village is a 'pixel'), while many presidential elections are similar to whole-picture voting.

Considering the simplest form of the proposed method presented in Section 2.3, in small dimension settings, we show below theoretical and experimental results to explain why our voting method outperforms the classical arithmetical voting method. More detail, when S_i is a scalar in $\mathbb{R}$ for all i, the two voting methods behave similarly, we shall prove that they both converge to θ with the same the rate of convergence $\sqrt{n}$. The difference can be seen when S_i is a vector and its components are dependent to describe the 'structure' inside it. We illustrate below the simulations for the cases when S_i is vector in $\mathbb{R}^2$ and its second component totally depends on its first component to show how our voting method behaves better than the arithmetical one.

6.1. One Dimension Case

In this case, $\{S_i\}_{i=1}^n$ are i.i.d. variables value in $\mathbb{R}$ with mean θ and variance σ. Recall that the classical soft arithmetical voting method (average voting) chooses $\Sigma_n = \frac{1}{n}\left(\sum_{i=1}^n S_i\right)$ as its final result, while our voting policy chooses

$$Y_n = \underset{S_i \in \{S_1, S_2, \ldots, S_n\}}{\arg\min} \sum_{j=1}^{n} d(S_i, S_j)$$

where d is some distance function. Here, we choose $d(x, y) = (x - y)^2$ to obtain a smooth function for analysis. The classical results show that Σ_n converges almost surely to θ (law of large number) and it converges in distribution to $\mathcal{N}(\theta, \sigma^2)$ with the rate of $\sqrt{n}$ (central limit theorem). With our voting method, we obtain the same guarantees for the convergence, as well as the rate of convergence, which are illustrated in Figure 11 and showed in Proposition 1. Remark that in Proposition 1, we restrict ourself by assuming that $\{S_i\}_i$ have positive, continuous, and bounded probability distribution in a neighborhood of the mean θ. However, this does “not restrict too much since most of the distributions (such as uniform distribution, normal distribution, exponential distribution, etc.) satisfy this restriction.

Proposition 1. *If $\{S_i\}_{i=1}^n$ have positive, continuous and bounded probability distribution in a neighborhood of θ, we have the following statements:*

1. $Y_n = \arg\min_{S_i \in \{S_1, S_2, \ldots, S_n\}} |S_i - \Sigma_n|$.
2. *Y_n converges almost surely to θ.*
3. *Y_n converges in distribution to $\mathcal{N}(\theta, \sigma^2)$ with the rate of $\sqrt{n}$.*

Proof. We will prove each statement in turn:

1. Define $h(x) = \sum_{j=1}^n (x - S_j)^2$. We have:

$$h'(x) = 0 \Leftrightarrow x = \frac{\sum_{j=1}^n S_j}{n} = \Sigma_n.$$

So we have:

x	$-\infty$		Σ_n		$+\infty$
$h'(x)$		$-$	0	$+$	
$h(x)$	$+\infty$	↘	$h(\Sigma_n)$	↗	$+\infty$

From variations of function h we imply that the one in $\{S_1, S_2, ..., S_n\}$ closest to Σ_n either from the left or the right is the arg min of h. On other hand, h is symmetric at Σ_n, i.e., for every ε, $h(\Sigma_n + \varepsilon) = h(\Sigma_n - \varepsilon)$. Indeed,
define

$$a_j = \frac{1}{n}\big(S_1 + S_2 + \cdots + S_{j-1} - (n-1)S_j + S_{j+1} + \cdots + S_n\big),$$

then it is obvious that $\sum_{j=1}^{n} a_j = 0$.
We have

$$h(\Sigma_n + \epsilon) = \sum_{j=1}^{n}(a_j + \epsilon)^2 = \sum_{j=1}^{n} a_j^2 + n\epsilon^2 + 2\epsilon\sum_{j=1}^{n} a_j = \sum_{j=1}^{n} a_j^2 + n\epsilon^2,$$

since $\sum_{j=1}^{n} a_j = 0$.
Similarly, we have

$$h(\Sigma_n - \epsilon) = \sum_{j=1}^{n}(a_j - \epsilon)^2 = \sum_{j=1}^{n} a_j^2 + n\epsilon^2 - 2\epsilon\sum_{j=1}^{n} a_j = \sum_{j=1}^{n} a_j^2 + n\epsilon^2.$$

So we obtain the first statement.

2. Denote $Z_n = Y_n - \Sigma_n$, we have Z_n converges almost surely to 0 and that thus implies Y_n converges almost surely to θ. Indeed,

$$\begin{aligned}
\mathbb{P}(|Z_n| \geq \varepsilon) &= \mathbb{P}(|Y_n - \Sigma_n| \geq \varepsilon) \\
&= \mathbb{P}\big(\cap_{k=1}^{n}\{|S_k - \Sigma_n| \geq \varepsilon)\}\big) \\
&\leq \mathbb{P}\big(\cap_{k=1}^{n}\{|S_k - \theta| \geq \varepsilon/2 \cup |\Sigma_n - \theta| \geq \varepsilon/2\}\big) \\
&\leq \mathbb{P}\big(\cap_{k=1}^{n}\{|S_k - \theta| \geq \epsilon/2\}\big) + \mathbb{P}(|\Sigma_n - \theta| \geq \epsilon/2) \\
&\leq q^n + 2\sigma\frac{\exp(-n\epsilon^2/8\sigma^2)}{\sqrt{2\pi n\epsilon}},
\end{aligned}$$

where $q = \mathbb{P}\{|S_k - \theta| \geq \epsilon/2\} < 1 \forall k = 1, ..., n$ for all $\epsilon > 0$ since by the assumption S_k has positive continuous distribution at θ, and Σ_n converges in distribution to $\mathcal{N}(\theta, \sigma^2/n)$ based on central limit theorem and from upper-tail inequality of normal distribution [24] stating that if $S \sim \mathcal{N}(0,1)$ then for every $x > 0$ we have

$$\mathbb{P}(X > x) \leq \frac{\exp(-x^2/2)}{x\sqrt{2\pi}}.$$

From theorem 7.5 in [25] we imply Z_n converges almost surely to 0 and therefore Y_n converges almost surely to θ.

3. The third statement is implied from the fact that Σ_n converges in distribution to $\mathcal{N}(\theta, \sigma^2)$ with the rate of $\sqrt{n}$ (central limit theorem) and Y_n (the element of $\{S_1, S_2, ..., S_n\}$ closest to Σ_n) converges in probability to Σ_n with the rate of n. Indeed, to simplify the proof, we assume $\theta = 0$, then for any $0 < \alpha < 1$ we have:

$$\begin{aligned}
\mathbb{P}(n^\alpha|Y_n - \Sigma_n| \geq \varepsilon) = \;& \mathbb{P}(n^\alpha|S_k - \Sigma_n| \geq \varepsilon \forall k = 1, 2, ...n) \\
& = \mathbb{P}(|\Sigma_n| \geq \delta)\mathbb{P}(n^\alpha|S_k - \Sigma_n| \geq \epsilon \forall k = 1, 2, .., n) \\
& + \mathbb{P}(|\Sigma_n| < \delta)\mathbb{P}(n^\alpha|S_k - \Sigma_n| \geq \epsilon \forall k = 1, 2, .., n) \\
& \leq \mathbb{P}(|\Sigma_n| \geq \delta) \\
& + \mathbb{P}(|\Sigma_n| < \delta)\mathbb{P}(n^\alpha|S_k - \Sigma_n| \geq \epsilon \forall k = 1, 2, .., n) \\
& = \mathbb{P}(|\Sigma_n| \geq \delta) \\
& + \int_{t=-\delta}^{t=\delta} g_n(t)\prod_{k=1}^{n}\big(1 - \int_{s=t-\epsilon n^{-\alpha}}^{s=t+\epsilon n^{-\alpha}} f(s)ds\big)dt
\end{aligned} \tag{13}$$

where f is density of S_k and g_n is density of Σ_n.

With the assumption, g_n is positive, continuous and bounded in a neighborhood of 0 as well because it is the density function of Σ_n which is the average sum of $\{S_k\}_{k=1}^n$. Let $\mathcal{U}$ is the open neighborhood of 0 in which g_n and f are positive, continuous and bounded. We first fix $\delta > 0$, such that $[-\delta, \delta]$ is in $\mathcal{U}$ then if ϵ is small enough we have $[-\delta - \epsilon, \delta + \epsilon]$ is in $\mathcal{U}$ as well. Therefore we have:

- $\Delta := \inf_{[-\delta-\epsilon, \delta+\epsilon]} f(x) > 0$;
- $A := \sup_{[-\delta,\delta]} g_n < +\infty$.

Thus the first component of (13) converges to 0 since Σ_n converges to 0 a.s; and the second component converges to 0 when ϵ goes to 0 since:

$$\begin{aligned}
&\int_{t=-\delta}^{t=\delta} g_n(t) \prod_{k=1}^{n} (1 - \int_{s=t-\epsilon n^{-\alpha}}^{s=t+\epsilon n^{-\alpha}} f(s) ds) dt \\
&\leq \int_{t=-\delta}^{t=\delta} g_n(t) \left(1 - 2\Delta\epsilon n^{-\alpha}\right)^n dt \\
&\leq 2\delta A \left(1 - 2\Delta\epsilon n^{-\alpha}\right)^n \\
&\sim 2\delta A \left(2\Delta\epsilon\right)^{n^{1-\alpha}} \\
&\longrightarrow 0 \text{ as } n \longrightarrow +\infty.
\end{aligned}$$

which ends the proof. □

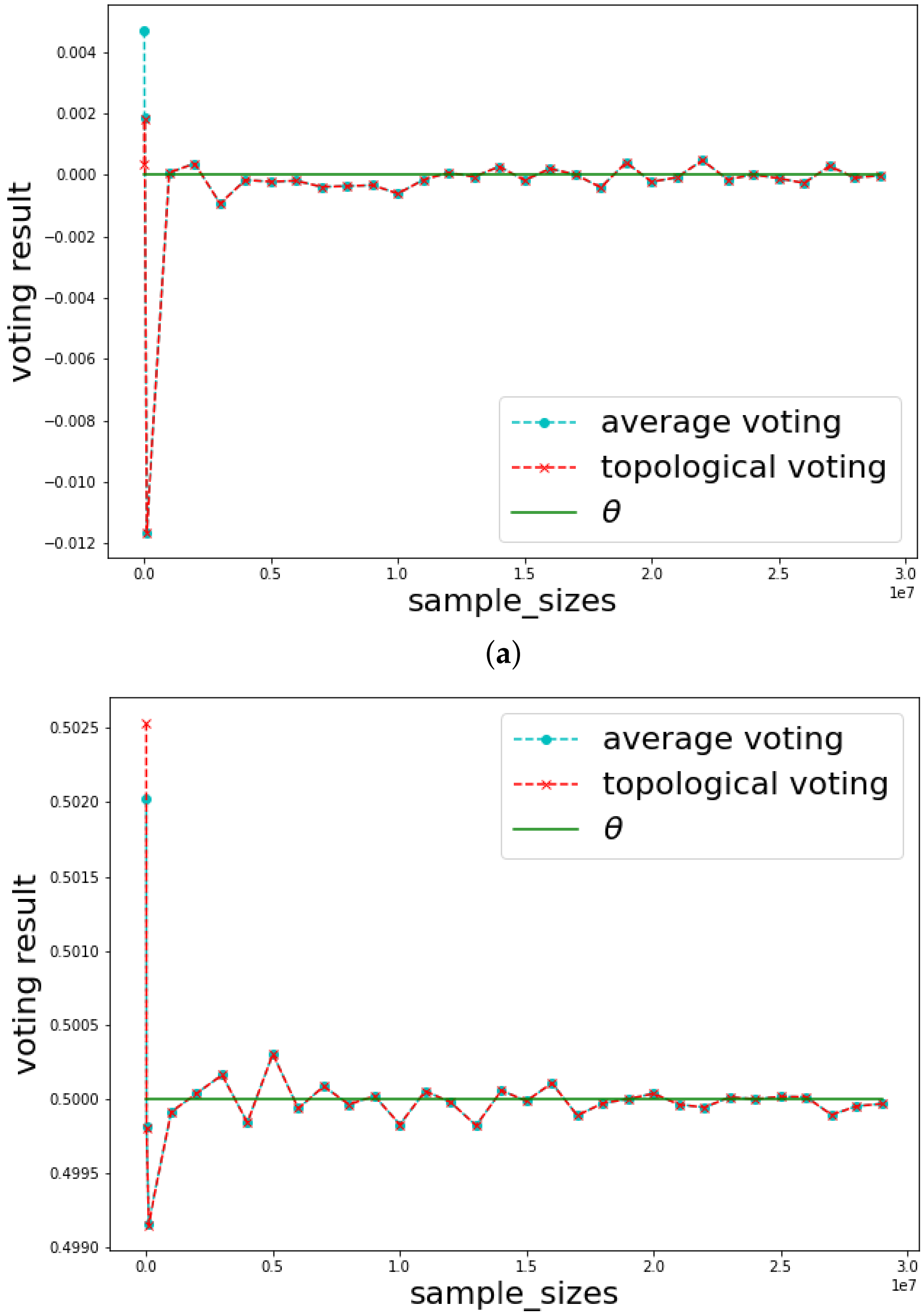
6.2. Two Dimension Case

Now, $\{S_i\}_{i=1}^n$ are vectors in $\mathbb{R}^2$. The masks in each nature segmentation problem have a natural "structure" in the meaning that its coordinates are not independent. For the illustrations below, we consider the simplest case where the second component of S_i totally depends on its first component, i.e., $S_i = [x_i, f(x_i)]$ for all i where $\{x_i\}_{i=1}^n$ are i.i.d. mean θ and variance σ and f are some functions.

In Figure 12, we vary function f to illustrate the different policies of the two voting methods. It can be seen that in the average voting, the first component converges perfectly to the true one but the second component seems to be very far away from the true one, the reason is simple, for most of function f, f of average is different from average of f. Meanwhile, the topological voting takes into account the trade-off between the two components.

In Figure 13, we take a symmetric function f. In this case, both of the two components of the result made by topological voting converge very well to the true one because of symmetry property of f.

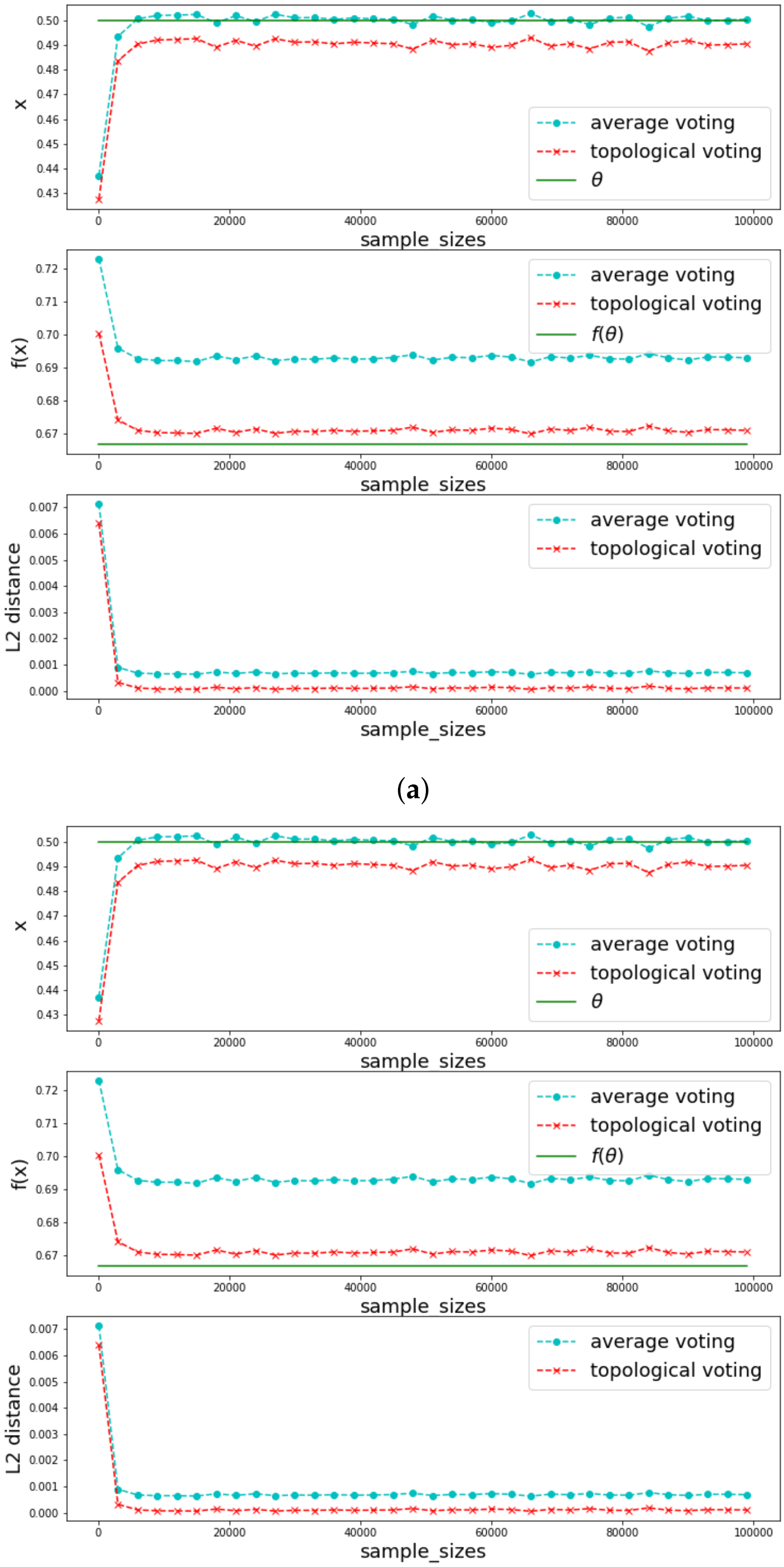
In Figure 14, we keep same f but vary distribution of x_i to see when the annotations are more or less variance how much different performance are between the two methods. One can see that when we decrease the uncertainty of the annotations, both of the two voting methods behave better especially for the topological voting. The improvement of topological voting to the arithmetical one is significant high when the uncertainty of the annotations is high.



(**a**)

(**b**)

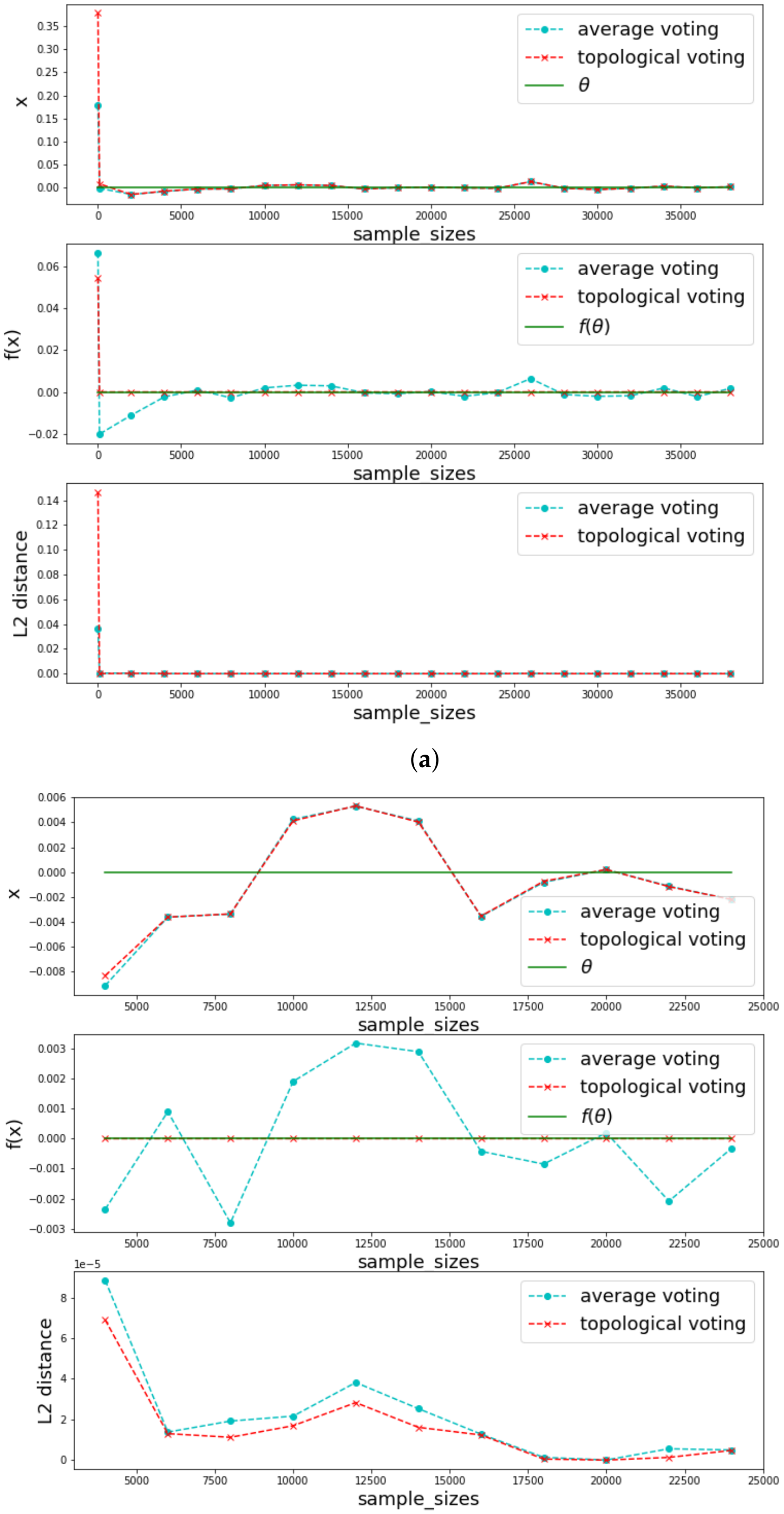
Figure 11. Comparisons of arithmetical voting (average voting) and topological voting method in one dimension setting. (**a**) $\{S_i\}_{i=1}^{n}$ are i.i.d. and follow $\mathcal{N}(0,1)$; (**b**) $\{S_i\}_{i=1}^{n}$ are i.i.d. and follow $\mathcal{U}(0,1)$.



(**a**)

(**b**)

Figure 12. Comparison in two-dimension case with same distribution of x_i but different f. (**a**) $x_i \sim \mathcal{U}(0,1)$ for all i, $f(x) = 1/(1+x)$; (**b**) $x_i \sim \mathcal{U}(0,1)$ for all i, $f(x) = x^2$.



(**a**)

(**b**)

Figure 13. Comparison when f is symmetric, $x_i \sim \mathcal{U}(-1,1), f(x) = x^3$. (**a**) Show for all samples; (**b**) Zoom in a segment of the samples.

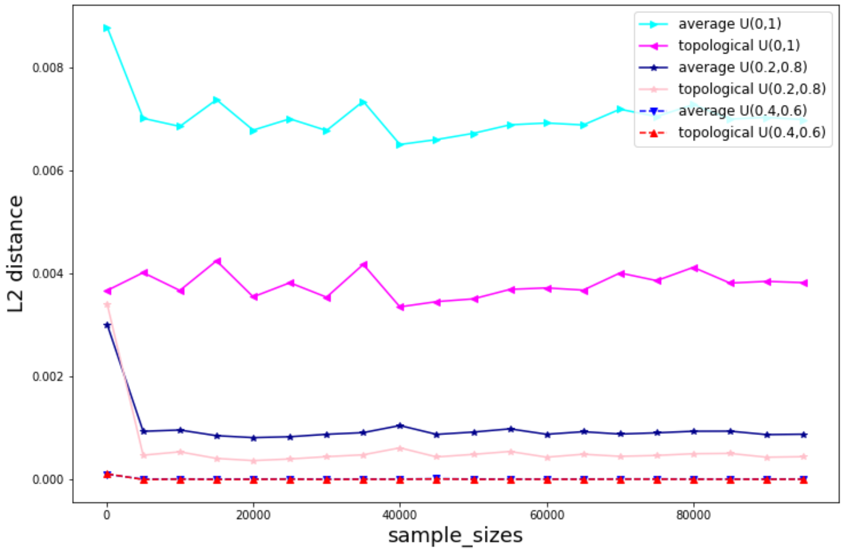


Figure 14. Comparison for different levels of uncertainty of the annotations x_i. $f(x) = x^2$.

7. Conclusions and Future Work

The paper considers various ensemble methods for image segmentation problems. We have presented the three proposed voting methods that take into account the structure of the whole mask. In three concrete examples, many experimental results are shown to prove that our voting methods outperform the classical arithmetical voting methods. We also provided some arguments about the philosophical reason, the mathematical guarantee and the experimental results (for the simplest version) to explain why the proposed topological voting methods behave better than the arithmetical one.

There are several directions in which this work can be taken. One avenue is to extend Proposition 1 for multiple dimension cases and to investigate the theoretical guarantee for the two other versions (the local and hybrid ones). Another avenue is to do more experimental comparisons with more other voting methods, for examples, the nine methods presented in [2].

Author Contributions: N.T.T.N.: Methodology, simulations, mathematical proofs and writing. P.B.L.: Statistic, writing, review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research is partially funded by Torus Actions SAS, Toulouse, France.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Exclude this statement.

Acknowledgments: The authors would like to thank very much Nguyen Tien Zung for posing to us the problem of topological voting, for suggesting many ideas on various versions of the voting method, and for his guidance during this work. The authors would like to thank very much Torus Actions SAS for supporting partially of this work.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Kim, H.; Thiagarajan, J.J.; Bremer, P.T. Image segmentation using consensus from hierarchical segmentation ensembles. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 3272–3276. [CrossRef]
2. Franek, L.; Abdala, D.D.; Vega-Pons, S.; Jiang, X. Image Segmentation Fusion Using General Ensemble Clustering Methods. In *Computer Vision—ACCV 2010*; Kimmel, R., Klette, R., Sugimoto, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 373–384.
3. Cyganek, B. One-Class Support Vector Ensembles for Image Segmentation and Classification. *J. Math. Imaging Vis.* **2011**, *42*, 103–117. [CrossRef]
4. Zhou, Z.H. *Ensemble Methods: Foundations and Algorithms*, 1st ed.; Chapman & Hall/CRC: Boca Raton, FL, USA, 2012.
5. Beyeler, M. *Machine Learning for OpenCV: Intelligent Image Processing with Python*; Packt Publishing Ltd.: London, UK, 2017.
6. Lam, L.; Suen, S.Y. Application of majority voting to pattern recognition: An analysis of its behavior and performance. *IEEE Trans. Syst. Man Cybern. Part Syst. Humans* **1997**, *27*, 553–568. [CrossRef]
7. Panagiotakis, C. Point Clustering via Voting Maximization. *J. Classif.* **2015**, *32*, 212 – 240. [CrossRef]
8. Buch, A.G.; Kiforenko, L.; Kraft, D. *Rotational Subgroup Voting and Pose Clustering for Robust 3D Object Recognition*; IEEE: Piscataway, NJ, USA, 2017; pp. 4137–4145.
9. Hu, Y.; Mageras, G.; Grossberg, M. Multi-class medical image segmentation using one-vs-rest graph cuts and majority voting. *J. Med. Imaging* **2021**, *32*, 034003. [CrossRef] [PubMed]
10. Jaccard, P. Distribution de la Flore Alpine dans le Bassin des Dranses et dans quelques régions voisines. *Bull. Soc. Vaudoise Des Sci. Nat.* **1901**, *37*, 241–272.
11. Rucklidge, W. *Efficient Visual Recognition Using the Hausdorff Distance*; Springer: Berlin/Heidelberg, Germany, 1996.
12. Rockafellar, R.; Wets, R. *Variational Analysis*; Springer: Berlin/Heidelberg, Germany, 2004; Volume 317. [CrossRef]
13. TGS. *TGS Salt Identification Challenge*. 2018. Available online: https://www.kaggle.com/c/tgs-salt-identification-challenge (accessed on 1 January 2022).
14. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
15. Chollet, F. Keras. 2015. Available online: https://github.com/fchollet/keras (accessed on 1 January 2022).
16. Milletari, F.; Navab, N.; Ahmadi, S.A. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016; pp. 565–571.
17. Phung, S.L.; Bouzerdoum, A.; Chai, D. Skin segmentation using color pixel classification: analysis and comparison. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 148–154. [CrossRef] [PubMed]
18. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv* **2019**, arXiv:1905.11946.
19. Rouchdy, Y.; Cohen, L.D. Geodesic voting methods: Overview, extensions and application to blood vessel segmentation. *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.* **2013**, *1*, 79–88. [CrossRef]
20. Malladi, R.; Sethian, J.A.; Vemuri, B.C. Shape modeling with front propagation: A level set approach. *IEEE Trans. Pattern Anal. Mach. Intell.* **1995**, *17*, 158–175. [CrossRef]
21. Chan, T.F.; Vese, L.A. Active contours without edges. *IEEE Trans. Image Process.* **2001**, *10*, 266–277. [CrossRef] [PubMed]
22. Udupa, J.K.; Samarasekera, S. Fuzzy Connectedness and Object Definition: Theory, Algorithms, and Applications in Image Segmentation. *Graph. Model. Image Process.* **1996**, *58*, 246–261. [CrossRef]
23. Staal, J.; Abramoff, M.; Niemeijer, M.; Viergever, M.; van Ginneken, B. Ridge based vessel segmentation in color images of the retina. *IEEE Trans. Med Imaging* **2004**, *23*, 501–509. [CrossRef] [PubMed]
24. Cardinal. Mathematics Stack Exchange. Version: 2011–03-24. Available online: https://math.stackexchange.com/questions/28751/proof-of-upper-tail-inequality-for-standard-normal-distribution/28754#28754 (accessed on 1 January 2022).
25. Pishro-Nik, H. *Introduction to Probability, Statistics, and Random Processes*; Lecture Notes; Kappa Research, LLC: Boston, MA, USA, 2014.