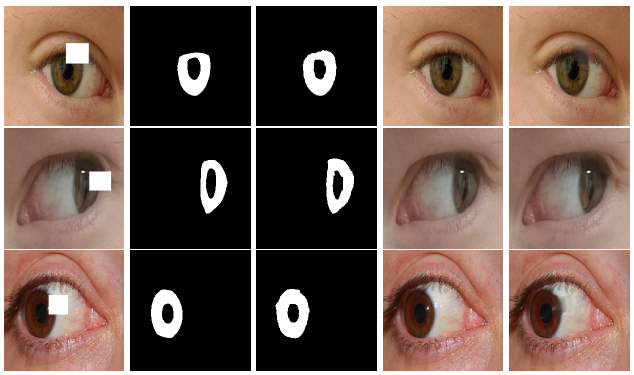
# Iris Segmentation Using Multi-Task Learning

Aljaž Đukić, Vitomir Štruc
University of Ljubljana
Tržaška cesta 25, 1000 Ljubljana, Slovenia
ad4794@student.uni-lj.si, vitomir.struc@fe.uni-lj.si

## Abstract

*The human iris is considered an extremely safe and reliable physiological modality and is thus often used in recognition systems. A crucial pre-processing step for quality iris recognition lies in iris segmentation, a process that determines which part of the captured image belongs to the iris. Iris segmentation has in recent years shifted from traditional algorithms to deep learning approaches, which have many advantages. In this paper, we follow this trend and with the use of multi-task learning, try to further improve the quality of iris segmentation in difficult settings and also its required training time. Our segmentation models follow the classic U-Net architecture, with certain modifications made to accommodate multi-task learning, where image inpainting was chosen as the auxiliary task. Our approach is evaluated on two datasets, MOBIUS and SBVPI, and shows promising results for training time improvements, albeit not for segmentation performance, where the results are improved only on the SBVPI dataset, but remain similar, yet worse on the MOBIUS dataset.*



(a) (b) (c) (d) (e)

Figure 1: Examples of the results of our multi-task iris segmentation learning approach, jointly trained with the task of image inpainting, for both tasks. The images in column (a) present the input images, those in columns (b) and (d) present the ground truths for both the main (iris segmentation) and auxiliary (image inpainting) tasks, and finally columns (c) and (e) display our results for each task.

## 1. Introduction

The human iris is considered to be an extremely reliable physiological modality, which is often used in biometrics for security and authentication systems [1]. Its advantage as a modality is most evident in its distinctiveness, permanence, and performance [7].

Iris segmentation is a process of determining which pixels of the input image belong to the iris and it presents a vital step for every sequential operation in the iris recognition pipeline (such as normalization and feature extraction) and is thus crucial for successful iris recognition. By segmenting the iris, we are removing irrelevant information, which could have a negative impact on the recognition results [10] and it has been proven [11] that the iris segmentation error rate negatively influences the quality of iris recognition. A quality segmentation step allows the extraction of more distinctive features and prevents invalid mapping of iris patterns into extracted iris code [7]. Furthermore, iris segmentation does not only play an important part in biometric recognition systems but is useful in medical diagnostic as well. Similar to iris recognition, the segmentation of the iris presents a crucial pre-processing step in complex computer systems designed for diagnostics of ocular diseases [7].

Iris segmentation has traditionally been solved with handcrafted methods, based on conventional computer vision, image processing, and pattern recognition methods. In the last decade, these traditional methods were surpassed by deep learning methods, which proved to be exceptionally good at solving different computer vision tasks, including image segmentation. Recent research in the field of iris segmentation focuses on segmentation in different difficult conditions such as with applications of recognition at-a-distance, on-the-move, with little or no user cooperation, using a mobile device, and recognition in dynamic

imaging environments. In these cases, the task of iris segmentation becomes increasingly challenging as opposed to in constrained environments with utilized user cooperation, where there are far fewer problems with noise, occlusions (due to eyelids, eyelashes, and glasses), spectral reflections, and off-angle image capturing. As further development of iris recognition systems aims to make iris recognition in difficult applications possible and increasingly reduce the required user cooperation [1], robust and quality iris segmentation is regarded as the first required step and remains an ongoing research topic [19].

While the deep learning methods used in these problematic applications achieve remarkable results in iris segmentation, they are all based on a single-task learning approach, where they focus only on iris segmentation. With this, some information that could be used to solve tasks related to segmentation is left out. These related tasks can share certain representations with the task of segmentation and by sharing them, a better generalization for all of the related tasks can be achieved [15]. Depending on the choice of the tasks, multi-task learning can either improve or worsen the results of all the tasks or only some of them. Because of this, it might be useful to choose multi-task learning even if solving only one task, known as the main task. With a good choice of an auxiliary task, a possible improvement of the main task, iris segmentation, can thus be achieved.

In this work, we try to address the gap in multi-task learning approaches for solving iris segmentation tasks. We propose a multi-task learning solution, to further improve the segmentation quality achieved with single-task learning. We also strive to improve on the time needed to train such models. Some examples of the results of our multi-task learning approach can be seen in Figure 1. The architecture we use for both single and multi-task learning models follows a classic U-Net architecture, where additional modifications are made to accommodate the multi-task learning approach.

The rest of our work is structured as follows. In Section 2 we present related works, where we look both into iris segmentation, as well as into image inpainting. We continue with our methodology explained in Section 3, followed by the experimental Section 4 where we present the datasets used, the experimental setup, evaluation methods we used, and finally the analysis of the results of our work. Section 5, where we summarize our work and future plans, concludes our paper.

## 2. Related Works

As our work focuses on a multi-task learning approach, we divide the related works section into two parts. We first present related works pertinent to our main task, iris segmentation, which is followed by related works of our auxiliary task, image inpainting.

### 2.1. Iris Segmentation

The development of iris segmentation algorithms started for the purpose of iris recognition. One of the first algorithms used for iris recognition was presented by Daugman [4]. With this, the development of traditional handcrafted iris segmentation methods, based on computer vision and image processing methods, began. Of course, the recent success of deep learning methods marked a turning point for iris segmentation as well. As such, iris segmentation algorithms can be broadly split into two categories: traditional approaches and deep learning algorithms.

**Traditional Methods:** The most common conventional iris segmentation methods are boundary based methods such as Daugman's integro-differential approach [4] and Wildes' [20] approach using Hough transform. Daugman's algorithm finds the inner and outer boundaries of the iris with the use of an integro-differential operator, while Wildes uses Hough transform following edge detection on a given image [1].

Pixel-based methods belong in the conventional method group as well. These methods determine the iris region using specific gradients for colour, texture, and illumination, thus separating the iris and non-iris region [1]. Among traditional iris segmentation approaches, there are also active contour methods [5], circle fitting methods [16], and others.

**Deep Learning:** In the past decade, deep learning methods have become an extremely popular and successful approach for solving many computer vision challenges, including iris segmentation and recognition [19].

Fully convolutional networks presented a big milestone for the challenge of image segmentation, allowing the problem of segmentation to be treated as a classification problem for each pixel of the image. Some of the approaches to iris segmentation using fully convolutional networks have been proposed by Liu *et al*. [9] and Bazrafkan *et al*. [3]. Their methods use modified fully convolutional networks for iris segmentation but lack fine enough results due to coarse upsampling of the feature maps [7]. In general, some fully convolutional architectures proved to be exceptionally good at segmentation challenges. Examples of these are Mask R-CNN [6], developed for general object segmentation, and U-Net [12], developed for segmentation of biomedical images, such as microscopical images of cells. U-Net and its variations specifically have proven to provide a good approach towards iris segmentation challenges. Lozej *et al*. [10] have shown that their U-Net based model outperforms multiple baseline methods for iris segmentation which use a combination of Hough transform and other image processing operations. Lian *et al*. [7] have upgraded the basic U-Net architecture to improve the wrongly classified,

noisy pixels outside of the iris region, by adding a regression module that helps the network focus on the iris region. The authors have shown that their approach improves the segmentation quality of the basic U-Net.

The popularity of deep learning based methods can also be observed in the recent iris segmentation challenges such as NIR Iris Challenge Evaluation in Non-cooperative Environments: Segmentation and Localization [19]. In this challenge, all of the participating teams had submitted algorithms based on deep learning methods.

### 2.2. Image Inpainting

The goal of image inpainting tasks is to fill out the missing pixels of the input image with coherent and accurate values [21]. The methods used for this could again be broadly split into conventional and deep learning methods.

Conventional methods often achieve image inpainting by propagating information about the missing pixels from the neighbouring ones. This approach is limited to small missing regions with little variance in colour and texture [8]. Inpainting is also done with patch-based methods, where the missing patches are filled with similar patches from the remaining image. Here the region with the best-calculated similarity is chosen. These methods work well on filling the simple background patches but fail at complex and non-repetitive regions [21]. Furthermore, calculating patch similarity is a computationally expensive process and even with additional upgrades, such as PatchMatch [2], these methods are still mostly unusable for real-time applications.

Deep learning models, unlike conventional methods, learn semantic priors and hidden representations. Usually defining a fixed value for missing pixels, these methods often achieve blurred results, with missing textures and colour discrepancies, where the results need to be additionally post-processed. Moreover, these methods often focus on missing regions of rectangular shape, which results in poor performance on irregular-sized patches [8]. Liu *et al*. [8] proposed a method that fixes the problems of region shape, dependence on starting pixel values and thus requires no post-processing. Their method uses partial convolutions, applying convolution only on valid pixels defined with a mask, and outperforms other noted compared methods.

## 3. Methodology

In this section, we present our proposed multi-task learning approach for the task of iris segmentation. We also describe the chosen architecture of our models, their modifications, and conclude with an explanation of the training process used in our work.

### 3.1. Overview of the proposed approach

Our iris segmentation approach is based on multi-task learning. This means that a single built model is capable of solving more than one task, an approach that can improve the speed of solving multiple tasks, the performance of some or all learnt tasks, and the training time needed to build such models. Our goal for choosing this approach is the improvement of the quality of the iris segmentation, which is therefore chosen as the main task. We chose image inpainting to be learnt alongside iris segmentation, as an auxiliary task. This way we try to achieve the sharing of certain joint representations between the chosen tasks, which could have a positive impact on the generalization of the model and thus the performance of our main task. As we focus only on the improvement of the iris segmentation, the performance of the auxiliary task is not important.

The proposed approach, seen in Figure 2 on the left, consists of modifying the chosen convolutional neural network architecture, defining the loss function, which includes the losses of both tasks, and their weights, which have to finally be fine-tuned, and additionally modifying the training data, so it can be used for multi-task learning with the chosen auxiliary task. The performance of this approach is compared to our baseline single-task learning model, trained only for the task of iris segmentation, using unmodified architecture and with unmodified training data.

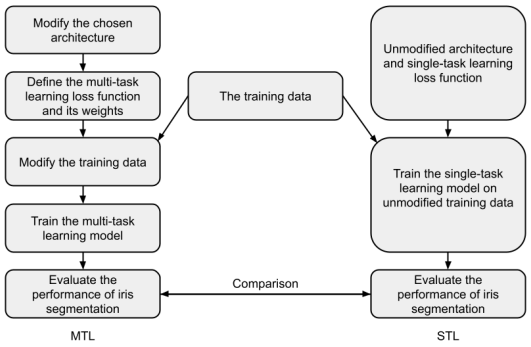



Figure 2: A step-by-step explanation of our methodology, with the multi-task learning approach shown on the left and the baseline single-task learning approach on the right.

### 3.2. U-Net overview

Both single-task and multi-task learning models, used in our work, are based on the classic U-Net architecture, proposed by Ronneberger *et al*. [12]. U-Net architecture is a commonly used approach for image segmentation, which was first developed for the segmentation of biomedical images. Its architecture is divided into two parts, an encoder, and a decoder. Encoder's primary role is capturing the broader context of the image, which is achieved with a sequential combination of convolutional and pooling layers, that down-sample the image and obtain its context. This process comes with the downside of worsening the localization accuracy, so a trade-off between the use of context

and the quality of localization is required. The authors of U-Net tackle this problem with an added decoder, which upsamples the intermediate feature maps, and combines them with corresponding high-resolution feature maps from the encoder. This way high-resolution results which still capture the broader context of the input image can be achieved [12].

### 3.3. Modifications to the architecture

The original U-Net's input and output sizes differ by a large margin of around $33\%$, which is caused by the use of unpadded convolutions that essentially crop the feature maps. We avoid such a discrepancy by applying padding for each convolution operation in the architecture. Furthermore, in the decoder's part of the architecture, we use transposed convolution layer for upsampling, instead of a combination of an upsampling operation and a $2 \times 2$ convolution. We also apply additional regularization layers, such as dropout layers, throughout the architecture instead of only at the end of the encoder.

To accommodate the architecture for multi-task learning, an additional output of the model is added, besides the existing segmentation output of the single-task learning.

### 3.4. Training

For the purpose of training a single-task learning model, the images of the datasets were not additionally modified. These RGB images represent the input data and iris masks represent their labels. However, for the part of multi-task learning, we had to additionally process them to be suitable for the task of image inpainting. For each input image, based on the corresponding iris masks, a white square of a random size was randomly added on the contour of one of the iris boundaries, thus removing a part of the image. Images processed in such a way can then be used as input images, with two corresponding labels. The first label presents the image before removal and is used to train inpainting, while the second label remains the iris segmentation mask used as before for the single-task learning approach.

The multi-task learning model is trained using a loss function, which contains loss functions of both outputs. Mean squared error is used for image inpainting while binary cross-entropy is used for the segmentation task. The total loss is calculated based on the values of each loss function and the loss weights defined beforehand. These weights ultimately influence the importance of each task and control by how much each of them will affect the learning process. Choosing suitable weights is tricky, and desired results require a lot of fine-tuning. Our final choice for weight values was $0.1$ and $1$ for mean squared error loss and binary cross-entropy loss, respectively.

Finally, when fitting both models, the ADAM optimizer is used with a default learning rate of $0.001$. We also use early stopping based on validation loss. Here we suppose that if the model has not improved in the chosen number of sequential epochs, it had probably converged close enough to its final value.

## 4. Experiments and results

In the following section, we present our experimental work, where we start with a description of the datasets that we used, followed by an explanation of our experimental setup and evaluation metrics. Finally, we conclude the section with the results of our experiments.

### 4.1. Datasets

To train and evaluate the quality of our approach, two datasets were used. These are MOBIUS and SBVPI datasets [18, 14, 13, 17], which are both publicly available and were both developed at the University of Ljubljana. The datasets were chosen because they contain quality segmentation masks of the iris region corresponding to the images of the dataset, which is necessary for developing and evaluating a deep learning iris segmentation model. The main characteristics of each dataset can be seen in Table 1 and we also provide a detailed description of each of them.

| Dataset | Total | Image details | Variations |
|---|---|---|---|
| MOBIUS | $16,717$ | $3000 \times 1700$, RGB | EV,CD,LC,GD |
| SBVPI | $1,858$ | $3000 \times 1700$, RGB | EV,GD |

Table 1: Overview of the datasets used. Variations of images are denoted as EV - eye variation, CD - capturing device, LC - lighting conditions, GD - gaze direction.

**MOBIUS dataset:** MOBIUS (the Mobile Ocular Biometrics in Unconstrained Settings) dataset consists of $16,717$ RGB colour images of eye regions taken from $100$ Caucasian individuals. The images were taken twice for each eye with three different mobile phones and under three different lighting conditions. Furthermore, the images differ in the direction of the individual's gaze, where four different gaze directions were photographed. For our work, only the segmentation subset was used. It consists of $3559$ hand-annotated RGB colour images of $35$ individuals. Each image in this subset includes corresponding segmentation masks of the sclera, iris, and pupil, but as we focused only on iris segmentation, sclera and pupil masks were not used.

**SBVPI dataset:** SBVPI (Sclera Blood Vessels, Periocular and Iris) dataset consists of $1,858$ RGB colour images of eye regions taken from $55$ Caucasian individuals. The images were taken four times for each eye with a digital camera, with no variation in the lighting conditions. Again four

Figure 3: Examples of the MOBIUS dataset images with the corresponding masks below them.

different gaze directions were used in each set. Similar to the MOBIUS dataset, a subset of 129 images are available with corresponding segmentation masks for the iris region.

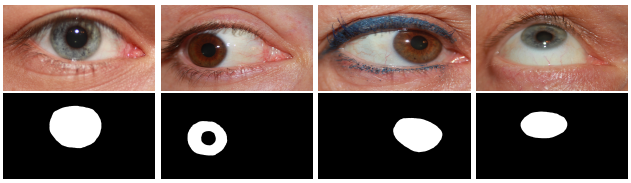


Figure 4: Examples of the SBVPI dataset images with the corresponding masks below them.

### 4.2. Experimental setup

The datasets used were first split into train and test subsets, with an 80% and 20% split. Furthermore, an additional 90% and 10% split of the training subset was applied, where 90% of the images were kept for training, and the rest of them were used for the validation subset to assess overfitting or underfitting of the model during training. Before splitting the data we also removed 10 images from the SBVPI dataset, where inconsistent annotations occurred. A random shuffle was also applied for each dataset, to thoroughly mix the images of all the individuals. The final number of images in each subset can be seen in Table 2.

| Dataset | Total | Train. subset | Val. subset | Test subset |
|---|---|---|---|---|
| MOBIUS | 3559 | 2562 | 285 | 712 |
| SBVPI | 119 | 85 | 10 | 24 |

Table 2: Overview of the segmentation subsets after splitting the images.

Before training the models we also resized the images and corresponding masks to the size of $320 \times 320$ and normalized their values between 0 and 1. Additionally, to reduce overfitting, we augmented the training subset before using it in the training process. Here we applied operations like horizontal and vertical flipping, zooming, shifting the image by its width or height, and also limited rotations.

For each dataset, multiple single-task and multi-task learning models were trained. The performance of the models in each category varies, specifically because of the training process, which is non-deterministic, and also because of our use of early stopping. By training multiple models of each type, calculating their performance on the test subset, and choosing the one with the best performance, we can determine which approach performs better with higher confidence. The same was done when evaluating training speed. Here instead, models were trained for a limited time, and again evaluated using the test subset and finally, the performance of the best in each category was compared, to determine which approach achieves better performance with the limited training time. It is important to note that even though the multi-task learning model is trained on modified images with rectangle holes added, the evaluation of both kinds of models is done on unmodified test images, as these are the images on which we try to improve iris segmentation.

### 4.3. Evaluation methods

To consistently evaluate the trained models we have chosen certain performance metrics, such as precision, recall, F1 score, and intersection over union metric, similar to [10]. Together these metrics can give us broader information about the performance of each evaluated model. They were chosen because they present a common way of interpreting the results of binary classification problems, such as image segmentation essentially is.

Each performance metric can be calculated for every test image, based on the resulting numbers of True Positive (denoted as TP), True Negative (denoted as TN), False Positive (denoted as FP), and False Negative (denoted as FN) classification outcomes for each pixel of the image. However, we report these metrics on the entire testing datasets, by summing the total number of outcomes across all the test images before calculating the results.

It is also important to note that the values of TP, TN, FP, FN, and consequently all of the performance metrics values are set with the chosen threshold. For each pixel classification, the model returns a probability pair for each category (iris or non-iris class) between 0 and 1. By choosing a threshold for iris classification we are thus influencing these values and performance metrics. For this reason, we report the threshold and its metrics where the best results are achieved.

Precision values tell us how many of our positive predictions were true, while recall measures how many of the total positive predictions we managed to predict. F1 score combines the two previous metrics and gives us better overall information about the performance. Intersection over union provides an insight into the overlap between our predictions and the ground truth. All of the used metrics can have val-

ues between 0 and 1, where a higher value is better.

Considering the outcomes of pixel classification, precision, recall and intersection over union (denoted as IoU) can be calculated with Equations (1), (2) and (3) respectively:

$$Precision = \frac{TP}{TP + FP}, \tag{1}$$

$$Recall = \frac{TP}{TP + FN}, \tag{2}$$

$$IoU = \frac{TP}{TP + FP + FN}. \tag{3}$$

Using the calculated values for precision and recall, we can also calculate the F1 score with Equation (4):

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}. \tag{4}$$

## 4.4. Results

We split our experimental work into two parts. In the first one, we evaluate and compare the performance of both single-task and multi-task learning approaches for the task of iris segmentation. The second part of the experiments focuses on their training speed, or how fast they converge towards their final values. Here the models were again trained and evaluated on the task of iris segmentation but had the number of their training epochs reduced to 10.

### 4.4.1 Performance comparison

In the first part, multiple single-task and multi-task learning models were trained. The performance of these models was evaluated for the task of iris segmentation using the corresponding test subset, on which the chosen metrics were calculated. The best performing model in each category and for each dataset was chosen for comparison, based on its highest F1 value (and also its intersection over union value). This way we were able to determine whether multi-task learning with the chosen auxiliary task of image inpainting improves the iris segmentation performance.

**Results:** The comparison of the best performing single-task and multi-task learning models for each dataset can be seen in Table 3, first for the MOBIUS dataset, followed by the results for the SBVPI dataset. The single-task learning model achieved better results of both the F1 metric and IoU metric for the MOBIUS datasets and can thus be determined as the better model for this dataset. It is worth noting that the performance of the multi-task learning model, though worse, is not much farther behind and still achieves quite similar results.

However, the multi-task learning model achieved better results for the SBVPI dataset, with a higher difference between the models compared to the MOBIUS dataset. This means that the results are inconclusive and we cannot confirm that multi-task learning with image inpainting as the auxiliary task improves the performance of the task of iris segmentation.

The cause for the differences in the results could be found in different ways of annotating the iris regions in each dataset. While MOBIUS' annotations cover only the iris region, SBVPI's annotations include the pupil as well. Another reason for the discrepancy could also lie in the fact that MOBIUS' images differ much more than SBVPI's, with many different lighting conditions and capturing devices covered in the dataset.

| Dataset | Model | Threshold | Precision | Recall | IoU | F1 |
|---|---|---|---|---|---|---|
| MOBIUS | STL | 0.45 | 0.8809 | 0.9171 | 0.8160 | 0.8987 |
| | MTL | 0.50 | 0.8775 | 0.9204 | 0.8156 | 0.8984 |
| SBVPI | STL | 0.50 | 0.9593 | 0.9667 | 0.9285 | 0.9629 |
| | MTL | 0.55 | 0.9695 | 0.9647 | 0.9362 | 0.9671 |

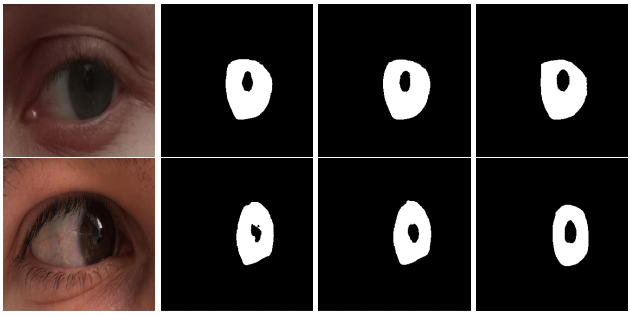
Table 3: Comparison between the best performing single-task (STL) and multi-task learning (MTL) models, first for the MOBIUS dataset, followed by the results for the SBVPI dataset.

**Qualitative evaluation:** A visual analysis of the results of both the single-task and multi-task learning models reveals extremely similar visual performance between the models of both datasets. This is supported by the performance results described above with similar performance metrics values between the models. Some examples of the segmentation results for MOBIUS and SBVPI test subsets can be seen in Figures 5 and 6 respectively, for both single-task and multi-task learning models. Visual inspection of the given segmentation results confirms quite accurate segmentation performance for both datasets.

### 4.4.2 Training speed comparison

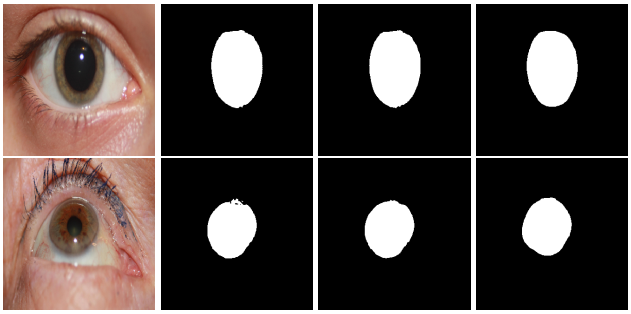
We continue with our experiments by looking at the training speed comparison between the single-task and multi-task learning models. To evaluate the differences between the models, we use the same approach as in the previous experiments, but limit the number of training epochs to only 10. By comparing the performance of the models after a limited training time we can determine if multi-task learning improves training speed. Here, we also provide a comparison between the values of loss functions during training, as another way of analyzing the training speed differences.

**Results:** The performance comparison of the best single-task and multi-task learning models, trained for 10 epochs,



(a) (b) (c) (d)

Figure 5: Some examples of the segmentation results for the MOBIUS dataset, with input images in column (a), single-task learning model's predictions in column (b), multi-task learning model's predictions in column (c), and the corresponding ground truth annotations in column (d).



(a) (b) (c) (d)

Figure 6: Some examples of the segmentation results for the SBVPI dataset, with input images in column (a), single-task learning model's predictions in column (b), multi-task learning model's predictions in column (c), and the corresponding ground truth annotations in column (d).

can be seen in Table 4, first for the MOBIUS dataset, followed by the results for the SBVPI dataset. It can be noted, that the multi-task learning model achieved better performance for both datasets when trained with limited training time. The difference between the models is slightly bigger for the SBVPI dataset. This could again be attributed to the differences in the datasets, namely the annotation style and image variations. Based on these results, it can be determined that multi-task learning with image inpainting chosen as the auxiliary task improves the training speed of the task of iris segmentation.

It is also interesting, that models trained on the SVBPI dataset achieved much worse results, even though their performance with non-limited training time far exceeded that of the MOBIUS dataset.

| Dataset | Model | Threshold | Precision | Recall | IoU | F1 |
|---|---|---|---|---|---|---|
| MOBIUS | STL | 0.45 | 0.8684 | 0.8867 | 0.7817 | 0.8775 |
| | MTL | 0.50 | 0.8601 | 0.9038 | 0.7880 | 0.8814 |
| SBVPI | STL | 0.70 | 0.7414 | 0.7854 | 0.6165 | 0.7628 |
| | MTL | 0.65 | 0.6727 | 0.8999 | 0.6258 | 0.7699 |

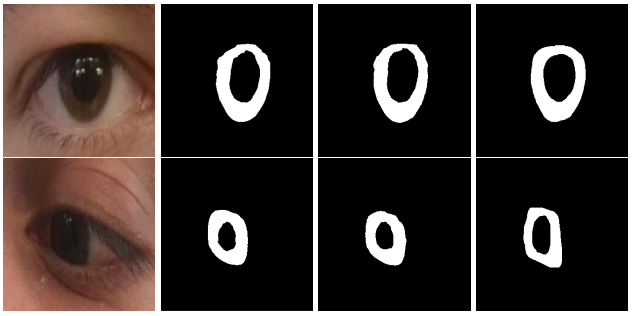
Table 4: Comparison between the best performing single-task (STL) and multi-task learning (MTL) models, that were trained for 10 epochs, first for the MOBIUS dataset, followed by the results for the SBVPI dataset.

**Qualitative evaluation:** Visual examination of the segmentation results for both datasets supports the performance results described above. While the values of precision and recall remain quite similar for both models of the MOBIUS dataset, this is not the case for the SBVPI dataset. Here, a large difference between the precision and recall values, yet similar IoU and F1, can be seen for the compared models, which is additionally confirmed with the visual inspection of the results. Some examples of the segmentation results for MOBIUS and SBVPI test subsets can be seen in Figures 8 and 9 respectively, for both single-task and multi-task learning models. It can be noted that both models of the MOBIUS dataset achieve visually similar results, while the SBVPI dataset's multi-task learning model predicts more pixels as the iris region than the single-task learning model. This way a larger portion of the True Positive outcomes is covered, thus reaching a higher recall value. However, this results in a lower precision quality, which is, in turn, decreased compared to the single-task learning model.

Furthermore, the visual quality of segmentation results is much worse for the SVBPI dataset models. This too is supported by the performance results described above, where the performance metrics values of the SBVPI models are far worse than those of the MOBIUS dataset.

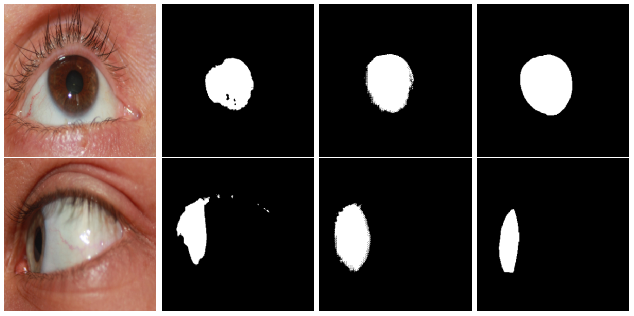
**Comparison of the loss function values:** Another way to compare the training speed of iris segmentation is to look at the values of loss functions during training. For this purpose, we compared the loss function values of the best performing single-task and multi-task learning models from the performance comparison of our experiments for both datasets. The comparison can be seen in Figure 7, where both models' loss functions are plotted on the same graph, first for the MOBIUS dataset, followed by SVBPI.

It can be noted that the loss function for multi-task learning reaches lower values sooner than the single-task learning loss function for both cases. As the loss function reflects the progress of the training, we can conclude that with the lower loss function values at the same epoch the training speed of the multi-task learning was faster and thus improved the training speed.



(a) (b) (c) (d)

Figure 8: Some examples of the segmentation results for the MOBIUS dataset, with input images in column (a), single-task learning model's predictions in column (b), multi-task learning model's predictions in column (c), and the corresponding ground truth annotations in column (d).



(a) (b) (c) (d)

Figure 9: Some examples of the segmentation results for the SBVPI dataset, with input images in column (a), single-task learning model's predictions in column (b), multi-task learning model's predictions in column (c), and the corresponding ground truth annotations in column (d).

## 5. Conclusion

We have examined the effect of the multi-task learning approach on the task of iris segmentation, with image inpainting chosen as an auxiliary task. The multi-task learning model achieved similar yet worse results to that of the single-task learning model, when trained and evaluated on the MOBIUS dataset, but improved the performance by a higher margin for the SBVPI dataset. This could be attributed to the differences in the datasets, namely in the annotation styles and image variation. Ultimately, the results of our performance experiments are inconclusive and cannot confirm that multi-task learning with image inpainting chosen as the auxiliary task increases the performance of iris segmentation.

However, multi-task learning did noticeably improve the performance of the models trained with a limited number of epochs. This results in a decrease of the required training time needed for the model's loss functions to reach values close to their final one. This could prove extremely useful when training models with very large datasets, where training times become increasingly longer. It could also prove useful when training a model with limited resources, such as on systems with poor computational power. In the future, we intend to look at the cause for the differences in the results of iris segmentation performance between the used datasets. We also intend to explore other possible auxiliary tasks for the proposed approach, such as image colorization and denoising and evaluate their effect on the iris segmentation performance as well as on the required training time.


Figure 7: Comparison between the loss function values for single-task (black) and multi-task (red) learning models, first for the MOBIUS dataset, followed by the values for the SBVPI dataset on the right.

## References


[1] M. Arsalan, H. G. Hong, R. A. Naqvi, M. B. Lee, M. C. Kim, D. S. Kim, C. S. Kim, and K. R. Park. Deep learning-based iris segmentation for iris recognition in visible light environment. *Symmetry*, 9(11):263, 2017.

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009.

[3] S. Bazrafkan, S. Thavalengal, and P. Corcoran. An end to end deep neural network for iris segmentation in unconstrained scenarios. *Neural Networks*, 106:79–95, 2018.

[4] G. Daugman, John G. (Huntingdon. Biometric personal identification system based on iris analysis. (5291560), March 1994.

[5] J. Daugman. New methods in iris recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(5):1167–1175, 2007.

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[7] S. Lian, Z. Luo, Z. Zhong, X. Lin, S. Su, and S. Li. Attention guided u-net for accurate iris segmentation. *Journal of Visual Communication and Image Representation*, 56:296–304, 2018.

[8] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018.

[9] N. Liu, H. Li, M. Zhang, J. Liu, Z. Sun, and T. Tan. Accurate iris segmentation in non-cooperative environments using fully convolutional networks. In *2016 International Conference on Biometrics (ICB)*, pages 1–8, 2016.

[10] J. Lozej, B. Meden, V. Struc, and P. Peer. End-to-end iris segmentation using u-net. In *2018 IEEE International Work Conference on Bioinspired Intelligence (IWOBI)*, pages 1–6, 2018.

[11] H. Proença and L. A. Alexandre. Iris recognition: Analysis of the error rates regarding the accuracy of the segmentation stage. *Image and vision computing*, 28(1):202–206, 2010.

[12] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[13] P. Rot, v. Emeršič, V. Štruc, and P. Peer. Deep multi-class eye segmentation for ocular biometrics. In *IEEE International Work Conference on Bioinspired Intelligence (IWOBI)*, pages 1–8, 07 2018.

[14] P. Rot, M. Vitek, K. Grm, v. Emeršič, P. Peer, and V. Štruc. Deep sclera segmentation and recognition. In A. Uhl, C. Busch, S. Marcel, and R. N. J. Veldhuis, editors, *Handbook of Vascular Biometrics*, pages 395–432. Springer, 2020.

[15] S. Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[16] S. Shah and A. Ross. Iris segmentation using geodesic active contours. *IEEE Transactions on Information Forensics and Security*, 4(4):824–836, 2009.

[17] M. Vitek, A. Das, Y. Pourcenoux, A. Missler, C. Paumier, S. Das, I. De Ghosh, D. R. Lucio, L. A. Zanlorensi Jr., D. Menotti, F. Boutros, N. Damer, J. H. Grebe, A. Kuijper, J. Hu, Y. He, C. Wang, H. Liu, Y. Wang, Z. Sun, D. Osorio-Roig, C. Rathgeb, C. Busch, J. Tapia Farias, A. Valenzuela, G. Zampoukis, L. Tsochatzidis, I. Pratikakis, S. Nathan, R. Suganya, V. Mehta, A. Dhall, K. Raja, G. Gupta, J. N. Khiarak, M. Akbari-Shahper, F. Jaryani, M. Asgari-Chenaghlu, R. Vyas, S. Dakshit, S. Dakshit, P. Peer, U. Pal, and V. Štruc. SSBC 2020: Sclera segmentation benchmarking competition in the mobile environment. In *IEEE International Joint Conference on Biometrics (IJCB)*, 10 2020.

[18] M. Vitek, P. Rot, V. Štruc, and P. Peer. A comprehensive investigation into sclera biometrics: A novel dataset and performance study. *Neural Computing & Applications*, pages 1–15, 2020.

[19] C. Wang, Y. Wang, K. Zhang, J. Muhammad, T. Lu, Q. Zhang, Q. Tian, Z. He, Z. Sun, Y. Zhang, T. Liu, W. Yang, D. Wu, Y. Liu, R. Zhou, H. Wu, H. Zhang, J. Wang, J. Wang, W. Xiong, X. Shi, S. Zeng, P. Li, H. Sun, J. Wang, J. Zhang, Q. Wang, H. Wu, X. Zhang, H. Li, Y. Chen, L. Chen, M. Zhang, Y. Sun, Z. Zhou, F. Boutros, N. Damer, A. Kuijper, J. Tapia, A. Valenzuela, C. Busch, G. Gupta, K. Raja, X. Wu, X. Li, J. Yang, H. Jing, X. Wang, B. Kong, Y. Yin, Q. Song, S. Lyu, S. Hu, L. Premk, M. Vitek, V. Štruc, P. Peer, J. N. Khiarak, F. Jaryani, S. S. Nasab, S. N. Moafinejad, Y. Amini, and M. Noshad. Nir iris challenge evaluation in non-cooperative environments: Segmentation and localization. In *2021 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–10, 2021.

[20] R. Wildes. Iris recognition: an emerging biometric technology. *Proceedings of the IEEE*, 85(9):1348–1363, 1997.

[21] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018.