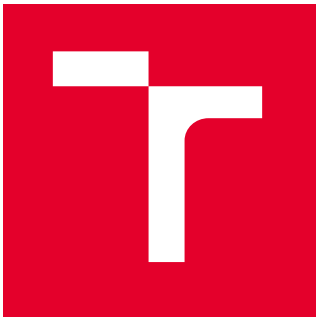


BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

REGISTRATION OF RETINAL IMAGES USING DEEP LEARNING

REGISTRACE SNÍMKŮ SÍTNICE POMOCÍ HLUBOKÉHO UČENÍ

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR — **Bc. Ondřej Doskočil**
AUTOR PRÁCE

SUPERVISOR — **Ing. Tomáš Vičar, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2023

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

Master's Thesis

Master's study program **Biomedical Engineering and Bioinformatics**

Department of Biomedical Engineering

Student: Bc. Ondřej Doskočil ***ID:*** 203657

Year of study: 2 ***Academic year:*** 2022/23

TITLE OF THESIS:

Registration of retinal images using deep learning

INSTRUCTION:

1) Through literature review, learn about the methods of registering image data using deep learning and perform a literature search of these methods. 2) Based on the literature search, select the most suitable method(s) for registering retinal images using deep learning and implement in the selected programming environment. 3) Perform basic testing on any suitable data. 4) Test and evaluate the success on the provided retinal images. 5) Optimise the method(s) appropriately for the data used. 6) Discuss the results obtained, focusing on comparisons with classical methods not using deep learning.

RECOMMENDED LITERATURE:

[1] FU Y, LEI Y, WANG T, CURRAN WJ, LIU T, YANG X. Deep learning in medical image registration: a review. Phys Med Biol. 2020 Oct 22;65(20):20TR01. doi: 10.1088/1361-6560/ab843e. PMID: 32217829; PMCID: PMC7759388.

[2] BOVERI HR, KHAYAMI R, JAVIDAN R, & MEHDIZADEH A. (2020). Medical image registration using deep neural networks: A comprehensive review. Computers & Electrical Engineering, 87, 106767.

Date of project specification: 6.2.2023 ***Deadline for submission:*** 22.5.2023

Supervisor: Ing. Tomáš Vičar, Ph.D.

prof. Ing. Valentine Provazník, Ph.D.
Chair of study program board

ABSTRACT

Image registration is a fundamental task in many medical image analysis applications. When examining the ocular fundus, it is necessary to register retinal images properly in order to examine phenomena such as vessel pulsations. This registration process is often solved using traditional, often iterative, methods. As in other areas of medical applications, deep learning methods have been increasingly used in recent years to achieve better and faster results. This thesis presents the possibilities of using deep neural networks to predict the transformation parameters to be used for retinal image registration.

KEYWORDS

image registration, deep learning, retinal images

ABSTRAKT

Registrace obrazu je zakladním úkolem v mnoha aplikacích pro analýzu medicínských obrazu. Při zkoumání očního pozadí je nutné dobře registrovat snímky sítnice, aby bylo možné zkoumat jevy jako pulzace cév. Tato registrace se často řeší pomocí tradičních, často iterativních, metod. Stejně jako v ostatních oblastech medicínských aplikací, se v posledních letech stále častěji používají metody hlubokého učení pro dosažení lepších a rychlejších výsledků. Tato práce prezentuje možnost využití hlubokých neuronových sítí pro predikci parametrů transformace, které se použijí pro registraci snímku sítnice.

KLÍČOVÁ SLOVA

registrace obrazů, hluboké učení, snímky sítnice

ROZŠÍŘENÝ ABSTRAKT

Úvod

Registrace snímků sítnice je zásadním krokem při vyšetřování očního pozadí. Při tomto vyšetření je možné sledovat například pulzace sítnicových cév, které můžou poskytnout cenné informace o různých stavech, jako je zvýšený nitrolební tlak, cévní obstrukce nebo glaukom. Hodnocení variability pulzace sítnicových cév umožňuje subjektivní hodnocení a hraje zásadní roli při včasné diagnostice onemocnění a předvídání jeho progrese. Díky možnosti včasné detekce a předvídání vývoje onemocnění lze vyvinout účinné terapeutické zásahy, které zabrání nevratné ztrátě zraku a zmírní dopady pozdních stadií onemocnění.

Předchozí studie v oblasti registrace snímků sítnice se často zaměřovaly na vývoj nástrojů využívajících konvenční metody, které pracují přímo s jasovými hodnotami, nebo ze snímků extrahují příznaky. Tyto metody však často zahrnují časově náročné iterační algoritmy. V posledních letech se jako slibné řešení složitých registračních problémů objevily techniky založené na hlubokém učení. Tyto techniky využívají hluboké neuronové sítě, které dokáží ze vstupních dat extrahovat komplexní příznaky a učit se z nich.

Tato práce se zaměřuje na použití metod hlubokého učení pro přímou predikci parametrů geometrické transformace. V teoretické části práce je provedena literární rešerše, která popisuje obecnou problematiku registrace obrazů a její implementaci pomocí tradičních metod. Dále jsou rozebrány různé přístupy registrace s využitím hlubokého učení. Z těchto přístupů je pak jeden vybrán a implementován do praxe. Na závěr je pak provedeno vyhodnocení a porovnání s tradičními metodami.

Popis řešení

Na základě rešerše byla vybrána metoda hlubokého učení, která je založena na predikci parametrů geometrické transformace. Jelikož se jedná o metodu, která využívá tzv. učení s učitelem, je nutné mít vstup a k němu jeho korespondující výstup. V případě predikovaných parametrů transformace je tedy nutné znát jaká transformace je mezi vstupními snímky. Toho bylo dosaženo pomocí náhodného generování transformačních parametrů a následnou transforamcí originálního snímku. Tímto postupem byla získána dvojice snímků a jejich odpovídající transformace, která je nutná pro správnou registraci. Takto připravená data jsou vhodná pro učení neuronové sítě.

Při návrhu samotné architektury sítě bylo vycházeno z tradičního postupu registrace obrazů. Architektura tedy obsahuje vrstvy, které napodobují jednotlivé kroky nutné k registraci. Architektura má podobu tzv. Siamské sítě, což je síť, která se

skládá z dvou ramen, které sdílí stejné parametry a váhy. Do každého ramena pak vstupují jednotlivé vstupy odděleně.

Celá architektura prošla několika optimalizacemi, které značně zlepšili výkonnost sítě. První stavební blok architektury je hluboká neuronová síť ResNet, která slouží pro extrakci příznaků. ResNet je jedna z nejpoužívanějších hlubokých neuronových sítí, která byla použita v řadě aplikací počítačového vidění, jako je třeba klasifikace obrazu, detekce objektů nebo segmentace. Dalším krokem v registraci obrazů je párování příznaků. Tento krok byl v úvodní architektuře implmentován v podobě konkatenace. Posledním krokem je predikce jednotlivých parametrů transformace z napárovaných příznaků. Tohle bylo implementováno jednoduše pomocí plně propojené vrstvy.

Ačkoliv takto navržená architektura fungovala, nedosahovala moc úctyhodných výsledků a proto byla provedena optimalizace. První blok architektury, který prošel optimalizací byl blok, který páruje příznaky, což bylo pomocí konkatenace. V literatuře se konkatenace často vyskytovala společně s jednoduchým odečtením. Implementace odečtení namísto konkatenace způsobila jen minimální zlepšení a proto byl implemetován jiný způsob. Tento způsob spočíval v korelaci. Korelační vrstva počítá korelaci mezi jednotlivými příznaky a výsledkem je podobnostní mapa, která dobře popisuje vztah mezi oběma vstupy. Implementace korelační vrstvy výrazně zlepšila celkovou výkonnost sítě a překonala konkatenaci a odečtení.

Pro další zlepšení byla optimalizována vrstva predikující jednotlivé parametry, která byla pouze ve forme plně propojené sítě. Tato vrstva byla nahrazena sofistikovanějším regresním blokem skládajícím se ze sekvence konvolučních vrstvev společně s batch normalizací a ReLu aktivační funkcí. Tato optimalizace ještě více přispěla k celkovému zlepšení a robustnosti celé architektury.

Vyhodnocení

Vyhodnocení bylo rozděleno do třech částí, kde každá hondnotí registraci jiným způsobem. V první části se provedlo vyhodnocení přesnosti predikovaných parametrů. Pro vyhodnocení byla použita průměrná absolutní chyba (MAE). Hodnocení porovnává přesnosti jednotlivých parametrů pro jednotlivé evoluce architektury. Jak bylo popsáno, korelační vrstva poměrně výrazně překonala konkatenaci a odečtení. Následně bylo potvrzeno, že implemetace regresního bloku zaznamenala velké zlepšení.

V druhé části vyhodnocení se hodnotila registrace za pomocí celých snímků. Byla počítána střední kvadratická odchylka (MSE) a strukturální podobnost (SSIM). Nejprve bylo provedeno vyhodnocení opět mezi jednotlivými evolucemi architektury, kde byl potvrzen závěr z minulé části. Následovalo porovnání finální architektury s tradičními metodami ORB a SIFT. Navržená architektura překonala v obou

metrikách tradiční metody.

V poslední části vyhodnocení se hodnotila stabilizace videosekvencí. Zde navržená architektura nedopadla nejlépe a byla překonána tradiční metodou SIFT. Hlavním důvodem bylo, že navržená architektura zanesla do registrovaných sekvencí nové chyby, které způsobily horší registraci. Tento problém byl hlavním nedostatkem navrženého řešení, jelikož síť predikovala jednotlivé parametry s rozdílnou přesností.

Celkově se dá však říci, že navržená architektura dokázala registrovat snímky sítnice, avšak je nutné ji více optimalizovat a přizpůsobit danému problému.

DOSKOČIL, Ondřej. *Registration of retinal images using deep learning*. Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Biomedical Engineering, 2023, 47 p. Master's Thesis. Advised by Ing. Tomáš Vičar, Ph.D.

Author's Declaration

Author:	Bc. Ondřej Doskočil
Author's ID:	203657
Paper type:	Master's Thesis
Academic year:	2022/23
Topic:	Registration of retinal images using deep learning

I declare that I have written this paper independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the paper and listed in the comprehensive bibliography at the end of the paper.

As the author, I furthermore declare that, with respect to the creation of this paper, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll. of the Czech Republic, Section 2, Head VI, Part 4.

Brno ....

author's signature*

*The author signs only in the printed version.

ACKNOWLEDGEMENT

I would like to thank the supervisor of my thesis, Ing. Tomáš Vičar Ph.D., for his professional guidance, consultation and expert suggestions in the elaboration of this thesis.

Contents

List of Figures

List of Tables

Introduction

Retinal vessel pulsation is a crucial diagnostic indicator that can provide valuable insights into various conditions, such as increased intracranial pressure, vascular obstruction, or glaucoma. Assessing the variability of retinal vessel pulsation allows for subjective evaluation and plays a vital role in early disease diagnosis and predicting disease progression. By enabling early detection and anticipating the advancement of conditions, effective therapeutic interventions can be developed to prevent irreversible vision loss and mitigate the impact of late-stage diseases.

The registration of retinal images is a critical step in ensuring accurate diagnosis. Previous studies in the field of retinal imaging have focused on developing registration tools utilizing conventional feature-based and intensity-based methods. However, these approaches often involve time-consuming iterative methods. In recent years, deep learning-based approaches have emerged as a promising solution for tackling complex registration problems, making them particularly well-suited for addressing the challenges in retinal image registration. Deep learning techniques leverage the power of neural networks to learn and extract meaningful features directly from the input data, eliminating the need for manual feature engineering. The utilization of deep learning in retinal image registration holds great promise for enhancing the accuracy and efficiency of the diagnostic process. By leveraging the capabilities of deep neural networks, we can expect improved registration performance and expedited workflows, ultimately leading to enhanced clinical decision-making and better patient outcomes.

This thesis focuses on the use of a supervised deep learning method for direct prediction of geometric transformation parameters. This supervision is achieved by generating synthetic images created by transforming the original retinal images with known affine transformation parameters. The proposed architecture of deep neural network has a Siamese network architecture and its individual parts mimics the traditional image registration approach with the difference that it is able to predict parameters in one pass.

1 Image registration

Image registration is a key step in a great variety of biomedical imaging applications. The aim of image registration is to create a spatially consistent pair of images (i.e. geometrically align one image with another). It is a prerequisite for all imaging applications that compare images across subjects, imaging modalities, or across time. Between individual images, the imaged scene may alter due to patient movements, physiological or pathological deformations of soft tissues, and so on. In order to work with such distorted images, it is necessary that images must be geometrically transformed. This requires finding a geometric transformation that aligns the scene image with the model image in a way that the transformed scene and the model are as similar as possible.

1.1 Geometrical transformations

Geometrical transformations involve manipulating the geometric properties of an image in a way that preserves its underlying structure. They are used to alter the position, size, shape, or orientation of an image in order to achieve a desired outcome. A geometrical transform of an image A is a mapping transformation T between its spatial coordinates $\boldsymbol{r} = (x, y)$ and the coordinates $\boldsymbol{r}' = (x', y')$ of the transformed image A' [17].

$$\boldsymbol{r}' = T(\boldsymbol{r}) \tag{1.1}$$

The values of the transformed image $f(r')$ are the same as the values of the original image $f(r)$, but in a different position [17].

$$f(\boldsymbol{r}') = f(T(\boldsymbol{r})) \tag{1.2}$$

Geometrical transformations can be divided into two basic groups, namely rigid and elastic [17]. Rigid transformations do not change the internal geometry of the image, but only shift or rotate the entire image. Elastic transformations, on the other hand, cause geometric deformation.

1.1.1 Rigid transformations

Rigid transformations are linear transformations that preserves length, and images after transformation are congruent (i.e. same size and shape). They only describe shift and rotation (Figure 1.1). The simplest example of a rigid transformation is a plain shift [17]. Plain shift moves every point of a image a constant distance in

a specified direction. This transformation has two parameters alongside individual coordinates $\Delta \boldsymbol{r} = [\Delta x, \Delta y]^T$ and can be written as

$$\boldsymbol{r}' = \boldsymbol{r} + \Delta \boldsymbol{r}. \tag{1.3}$$

Rotation,

$$\boldsymbol{r}' = \boldsymbol{B}\boldsymbol{r}, \tag{1.4}$$

is defined by rotation matrix $\boldsymbol{B}$ which causes the points in the two-dimensional space to rotate by the angle θ around the origin of the coordinate system [17], where

$$\boldsymbol{B} = \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix} \tag{1.5}$$

The generic rigid transformation contains both of these components, i.e. shift and rotation, and can be written as follows

$$\boldsymbol{r}' = \boldsymbol{B}\boldsymbol{r} + \Delta \boldsymbol{r}. \tag{1.6}$$

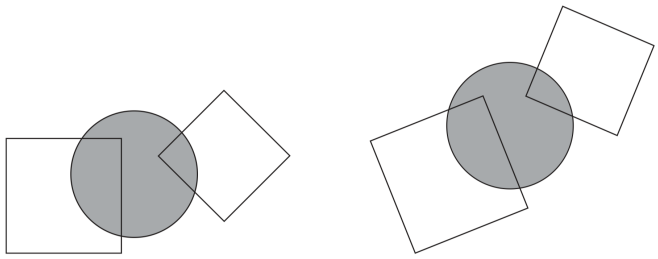


Fig. 1.1: Rigid transformation: (left) original and (right) shifted and rotated image, taken from [17].

1.1.2 Flexible transformations

Flexible transformations involve both changes in the shape and position of the image. They can be thought of as an image printed on a flexible substrate, parts of which can be stretched or compressed. The simplest transformation of this kind is a plain scaling

$$\boldsymbol{r}' = \boldsymbol{S}\boldsymbol{r} \tag{1.7}$$

where $\boldsymbol{S}$ is scaling matrix in the two-dimensional space defined as

$$\boldsymbol{S} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{1.8}$$

Values s_i can be either identical or different. If they are identical, scaling is isotropic (i.e. magnification or reduction). If values are different, proportions in the image will be changed (see Figure 1.2).

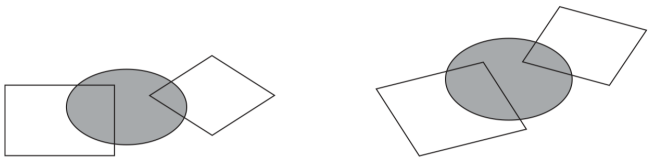


Fig. 1.2: Plain scaling of previous images from Figure 1.1 with differing values s_x and s_y, taken from [17].

Another flexible transformation is shearing (Figure 1.3), which can be defined in two-dimensional space as

$$\boldsymbol{G} = \boldsymbol{G}_x \boldsymbol{G}_y, \tag{1.9}$$

$$\boldsymbol{G}_x = \begin{pmatrix} 1 & g_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \boldsymbol{G}_y = \begin{pmatrix} 1 & 0 & 0 \\ g_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{1.10}$$

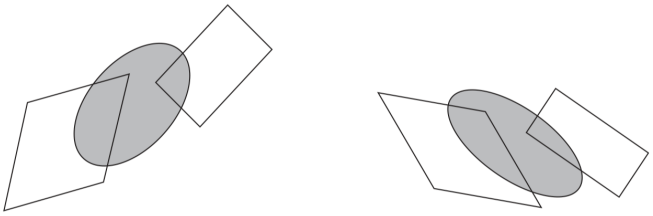


Fig. 1.3: Shearing: with differing values g_x, g_y, taken from [17].

All of these transformations can be combined into one generic linear transformation called affine transformation. This transformation contains translational-rotational matrix, scaling and shearing and is defined as

$$\boldsymbol{r}' = \boldsymbol{GSRr} = \boldsymbol{Ar} \tag{1.11}$$

where matrix $\boldsymbol{A}$ is in the two-dimensional space defined as

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & t_1 \\ a_{21} & a_{22} & t_2 \\ 0 & 0 & 1 \end{pmatrix} \tag{1.12}$$

1.2 Similarity metrics

Similarity metrics are utilized to evaluate the similarity between one or more images or their areas. The similarity metric can be based on different image properties, such as intensity, texture, or landmarks, and it can be defined in different domains, such as spatial, frequency, or wavelet [17] [30]. We can divide them into two categories: intensity-based and feature-based. Generally, intensity-based measures, such as Mean Square Distance (MSD) or sum-of-square distance (SSD), consider a complete mechanical correspondence between the given images. On the other hand, feature-based measures, such as Mutual Information [22], which is based on the information theory, searches for a structural correspondence between images [3]. It is also worth mentioning other frequently used methods as Cross-correlation (CC) and Dice Similarity Coefficient (DSC) [5].

1.3 Finding parameters of geometric transformation

After selecting the appropriate registering transform and similarity metric that best describes the problem to be solved, it is then necessary to find particular parameters of this transform. This is usually solved using optimization algorithms shown in the Figure 1.4. It is iterative process where optimal vector $\boldsymbol{\alpha}_0$ of transform parameters is find. This process can be written as optimization

$$\boldsymbol{\alpha}_0 = \arg \underset{\boldsymbol{\alpha}}{max}\, c(B(\boldsymbol{x}_B), A'(T_{\boldsymbol{\alpha}}(\boldsymbol{x}_A))),\ \boldsymbol{x}_B, T_{\boldsymbol{\alpha}}(\boldsymbol{x}_A) \in \boldsymbol{\Omega}_{\boldsymbol{\alpha}}. \tag{1.13}$$

That means that registering transform $T_{\boldsymbol{\alpha}}$ with parameter vector $\boldsymbol{\alpha}$ transforms image A into A' so it can be computed similarity metric c with base image B in overlapping area $\Omega_{\boldsymbol{\alpha}}$ [17].

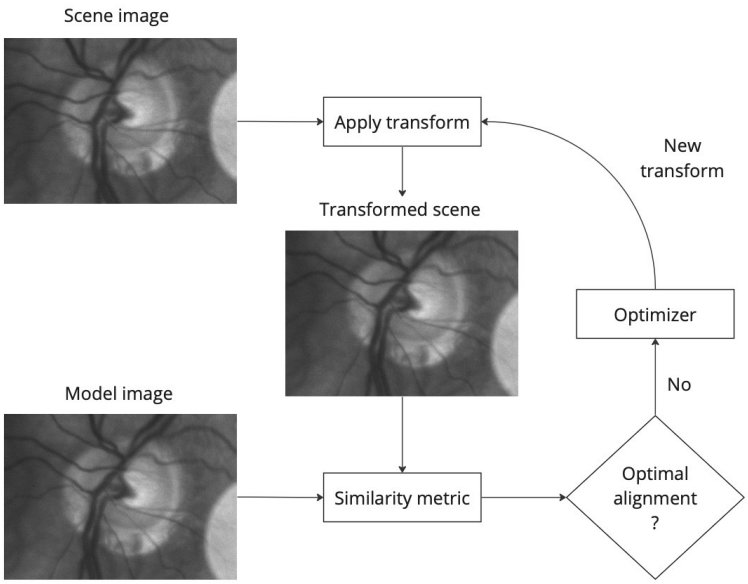


Fig. 1.4: General optimization algorithm of image registration

2 Conventional approaches for image registration

The variety of image registration data makes it very difficult to design an algorithm that is suitable for every occasion. This fact has given rise to a large number of methods for image registration. Since all methods do the basically the same thing, the majority of them consists of the following four steps: [30]

- Feature detection
- Feature matching
- Transformation model estimation
- Image transformation

Feature detection

Feature detection is the process of identifying and extracting distinctive features (such as corners, edges, and blobs) from an image that are likely to be present in both images. These features serve as the basis for determining the similarity between the images and, consequently, the required transformation [30]. There are several well-known techniques for feature detection, including:

- Harris Corner Detection: This method identifies corners in an image by looking for regions with large variations in intensity in all directions. This approach is robust, as corners are less likely to be altered by changes in lighting, rotation, or scale [24].
- SIFT (Scale-Invariant Feature Transform): SIFT is a popular feature detection algorithm that identifies and describes local features in images. It is highly robust and invariant to image scaling, rotation, and changes in illumination [21].
- SURF (Speeded Up Robust Features): SURF is an enhanced version of SIFT that aims to improve the speed and performance of feature detection. It uses a different approach to identify key points and compute descriptors, making it faster and more efficient than SIFT while maintaining similar levels of robustness. By focusing on feature detection rather than working with every pixel, the computational effort required for image registration is significantly reduced [2].

Feature matching

After detecting features in both images, the next step is to establish correspondences between these features. This involves finding similarities between the features in the two images and determining which pairs of features correspond to the same real-world object or point [30]. Some common methods for feature matching include:

- Nearest-neighbor matching: In this method, the distance between the descriptors of features in the two images is computed, and the closest pair is considered a match. This approach is simple and fast but may be prone to false matches [1].
- Random Sample Consensus (RANSAC): RANSAC is a robust method for finding the best set of matches by iteratively selecting random subsets of feature pairs, estimating the transformation model, and identifying inliers (correct matches) and outliers (false matches) [11].
- Clustering: This technique groups similar features together based on their descriptors. It helps in identifying and matching features that belong to the same object or region in the images [4].

Transformation model estimation

Once the correspondences between the features in the two images are established, the transformation model that maps one image onto the other can be constructed. It is important to choose the appropriate mapping function, which should correspond to the assumed geometric deformation of the sensed image. Common mapping functions include affine, projective, or polynomial transformations [30].

Image transformation

Lastly, after the transformation model has been estimated, one image can be transformed to align with the other. This step involves applying the estimated transformation model to the pixels in one image to obtain a transformed image that is aligned with the other. Various interpolation methods can be used to fill in the gaps and create a seamless transformation. Common interpolation methods include nearest-neighbor, bilinear, and cubic interpolation [30].

3 Deep learning approaches for image registration

There are many deep learning based methods that can be used to image registration and each approaches registration in a different way. As shown in Figure 3.1 they can be divided into three parent classes [5]. First category use neural networks as a similarity metric (often called deep-similarity) [29] [6]. These methods are usually employed for multi-modal image registration, due to the substantial variation in the appearance and intensity distributions of the moving and fixed images [13]. Second there are methods which predict parameters of the transformation model. Lastly there are methods that uses neural networks to extract features [28] or learn new image representations [20] to transfer the original image to new image which is more convenient for image registration [5].

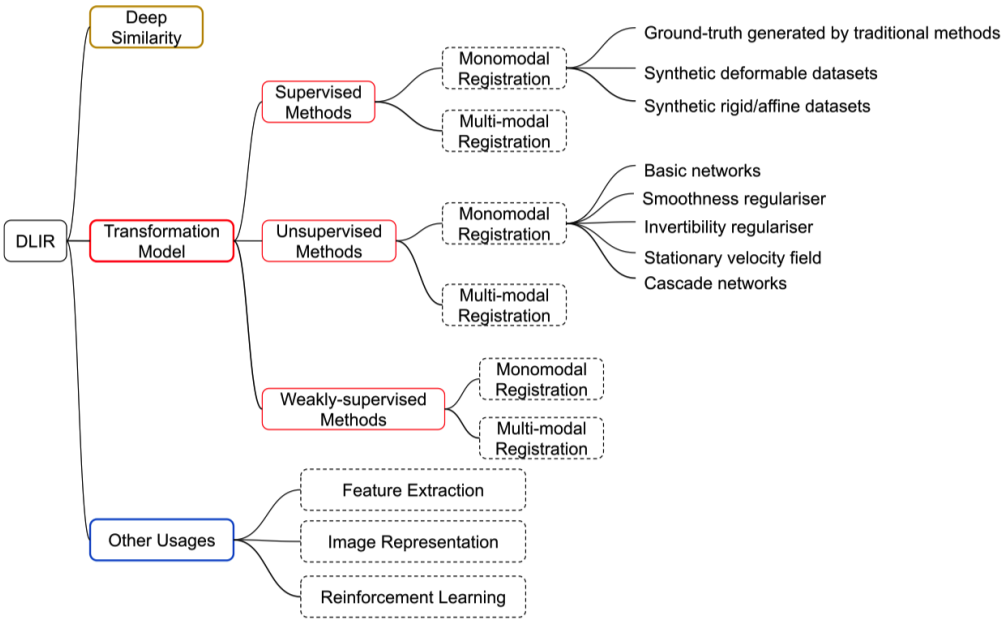


Fig. 3.1: Categories of deep learning-based image registration methods, taken from [5]

3.1 Deep similarity

Deep similarity is approach to use deep neural networks to act as similarity metric that can be used to compare the similarity of two images [28]. Traditional image registration techniques often rely on handcrafted features, which may not be robust enough to handle variations in image appearance due to changes in lighting, viewpoint, and other factors. In contrast, deep neural networks can learn features that are invariant to these variations and can capture more complex patterns in images. Especially for multi-modal registrations, deep similarity has proven to be very effective [13]. In general, these metrics usually outperform traditional metrics such as Mutual Information, but since they are only replacement for similarity metric throughout the image registration process, it is still iterative process [3].

3.2 Transformation model prediction

Transformation model prediction refers to the process of predicting parameters of the geometric transformation. Based on the machine learning paradigm used to train the networks, this category of approaches can be further divided into supervised, weakly-supervised, and unsupervised approaches, as shown in Figure 3.1. The primary benefit of this group of techniques over traditional approaches and deep similarity networks is the significant speedup they provide during inference, allowing for real-time rigid and non-rigid image registration . This speedup is achieved by predicting the geometric transformation in one pass [12] [5].

3.2.1 Supervised methods

Supervised methods for transformation model prediction involve deep neural networks that estimate spatial transformation parameters in a supervised fashion, using ground-truth values to guide the learning process. These methods thus rely on the availability of ground-truth data for the transformation parameters, which can be obtained through traditional registration methods or by using simulated images with known ground-truth transformations. Since estimating ground truths with traditional methods basically leads to creating network that performs similarly to used traditional method, it is better to predict parameters with known transformation using synthetic images. Also random combinations of operations such as rotation, translation, and scaling would suffice to provide the data required to train a network, making the ground truth for registration considerably easier to synthesise [5].

3.2.2 Unsupervised methods

Unsupervised deep learning methods for image registration do not require ground-truth transformations for training, addressing a significant limitation of supervised image registration methods. These methods often use convolutional neural networks (CNNs) combined with spatial transformer networks (STNs) to generate deformation fields that warp the moving image to match the fixed image. The dissimilarity between the warped moving image and the fixed image can be used to calculate the loss function for backpropagation. Several unsupervised deep learning approaches have been proposed to improve the smoothness and invertibility of the estimated deformation fields, leading to better registration performance [5].

3.2.3 Weakly supervised methods

Weakly-supervised deep learning methods involves the process of aligning images, with the help of additional information during training. This additional information, usually in the form of region-wise labels, masks, or landmarks, is used to preserve anatomical coherence between tissue or organ boundaries and guide the estimation of spatial transformations. The main idea is to optimize a loss function that matches both labels and images, ultimately estimating the desired deformation field. Unlike supervised methods that require ground truth deformation fields, weakly-supervised methods offer a more practical approach for real-world applications [5].

3.3 Other methods

In medical image registration, deep neural networks have been used for various purposes beyond predicting similarity metrics and transformation fields. These purposes include feature extraction, learning new image representations, reinforcement learning, and more [12].

Feature extraction methods capitalize on the ability of deep learning networks to efficiently extract meaningful features from images. By integrating these features with traditional registration techniques, researchers have achieved superior performance compared to conventional methods. These approaches have been particularly useful in multi-modal registration, where automatically learned intrinsic features have led to lower target registration error rates [5].

Image representation approaches, on the other hand, address the issue of low-quality or multi-modal images by generating new image representations with more distinguishable anatomical features. This ensures high registration accuracy even when the original images are of low quality or from different modalities. Some

studies have employed deep learning networks to learn mappings between modalities, transforming multi-modal registration tasks into mono-modal ones. Others have constructed shared spaces for images from different modalities [5].

Reinforcement learning has emerged as a promising approach in medical image registration, especially in the context of rigid registration tasks. In reinforcement learning-based image registration, the registration process is often described as a Markov Decision Process (MDP), which consists of states, actions, and rewards. The states represent the current alignment of the moving and fixed images, the actions correspond to possible transformations that change the alignment, and the rewards are based on a measure of alignment quality (e.g., similarity metric or registration error). The main objective of the reinforcement learning agent is to learn a policy that maximizes the cumulative reward over a sequence of actions, leading to an optimal registration [12] [5].

Overall, deep neural networks have a lot of potential in medical image registration and can be used for various purposes to improve the accuracy and efficiency of registration methods.

4 Design and implementation

Based on the research of related papers, supervised method for predicting transformation model parameters was selected. The primary reason for this choice is that supervised learning is considered to be the most straightforward approach. It involves training a model on a labeled dataset, where each sample is associated with a ground truth transformation model parameters. By optimizing the model to minimize the difference between its predictions and the ground truth labels, it can learn to accurately predict transformation parameters for new, unseen data.

4.1 Dataset

To train supervised network it is necessary to have ground truth values. These values were created using synthetic images with known transformation parameters. Given dataset of videos from video ophthalmoscope consists of videos from 78 patients. For each patient, there is one or more videos of different lengths for the left and right eye. These videos have been converted to single frame sequences. Each patient has approximately 1600 retinal images. And since the images in each video are very similar, 500 images were randomly selected for each patient. Finally, the patients were divided into training and testing sets in a ratio of 8/2, i.e. 62 patients for training and 16 patients for testing. Random samples are shown in Figure 4.1.

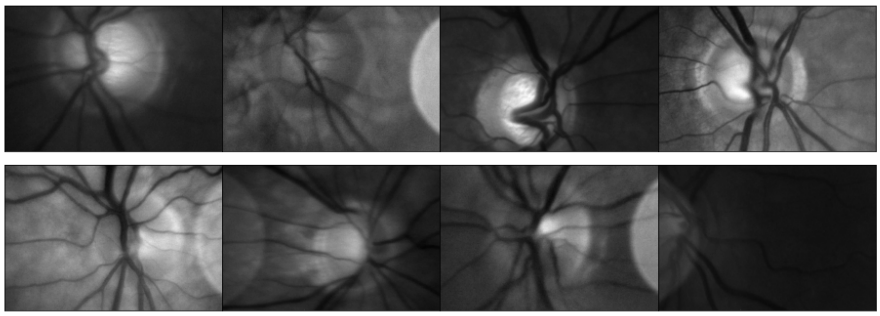


Fig. 4.1: Random samples of retinal images from given dataset

For each image, individual geometric transformation parameters (translation, rotation, scale and shear) were generated and converted into affine transformation parameters according to 1.11. These parameters were then used to transform the

original image, resulting in an synthetic transformed image with known transformation parameters. Data prepared in this way are ready to supervise learning of neural network as shown in Figure 4.3.

4.2 Network architecture

Choosing the right architecture for a deep neural network (DNN) is crucial for achieving accurate and efficient image registration. One promising architecture for this task is the Siamese network, which is designed to learn similarity between pairs of inputs.

4.2.1 Siamese networks

A Siamese neural network is a type of deep learning architecture that consists of two identical subnetworks, as shown in Figure 4.2, that share the same parameters and weights. Siamese network is commonly used for tasks that require comparing or matching two inputs, such as signature verification [9], face recognition [27], object tracking [14], and even natural language processing [23]. For example, in signature verification, the Siamese network can learn to extract features from two different signatures and predict whether they belong to the same person or not. The output of each subnetwork is combined by a similarity function or a distance metric, which measures the similarity between the two inputs. This metric can be used to determine whether the two inputs are similar or different, depending on the task at hand. The similarity function can be a simple distance metric, such as Euclidean distance or cosine similarity, or a more complex function, such as a neural network [9] [7].

Overall, the Siamese network is a powerful and flexible architecture that can be adapted to a wide range of tasks that involve comparing or matching two inputs. In this thesis, this architecture will be used and modified to predict parameters of geometric transformation on its output.

4.2.2 Initial architecture

As mentioned in previous chapter, Siamese networks are a good choice for transformation model prediction. In the creation of this architecture, the focus of this thesis was to make the overall architecture mimic steps of traditional way of image registration mentioned in Chapter 2, but using deep learning. ResNet [15], one of the most widely used deep neural networks, was used to extract features from the image. ResNet has been widely adopted in various computer vision tasks, such

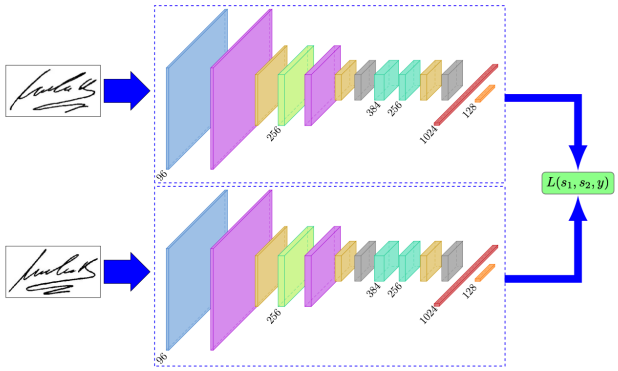


Fig. 4.2: Siamese network, used for signature verification. A different signature frame enters each branch, and at the output the network produces a similarity metric (contrastive loss in this case), edited from [9].

as image classification, object detection, and segmentation, and has consistently achieved state-of-the-art results. One of the reasons for its success is its ability to extract meaningful features from input data. ResNet's residual blocks allow the network to learn more abstract and complex features as the depth increases, which is especially useful for tasks that require understanding high-level concepts in the input data [15].

Specifically, the Resnet18 variant was used here. As the name suggests, the network has 18 layers and several residual blocks (the last fully connected layer has been omitted). Both outputs from the last ResNet's convolutional layer were simply concatenated. This concatenation act as feature matching step which is used several times in related papers (e.g. [8]). Then a fully connected layer is used for transformation model prediction, which is the last layer of the Siamese network and has an output dimension corresponding to the number of particular geometric transformation parameters i.e. 7. The whole architecture can be seen in Figure 4.3.

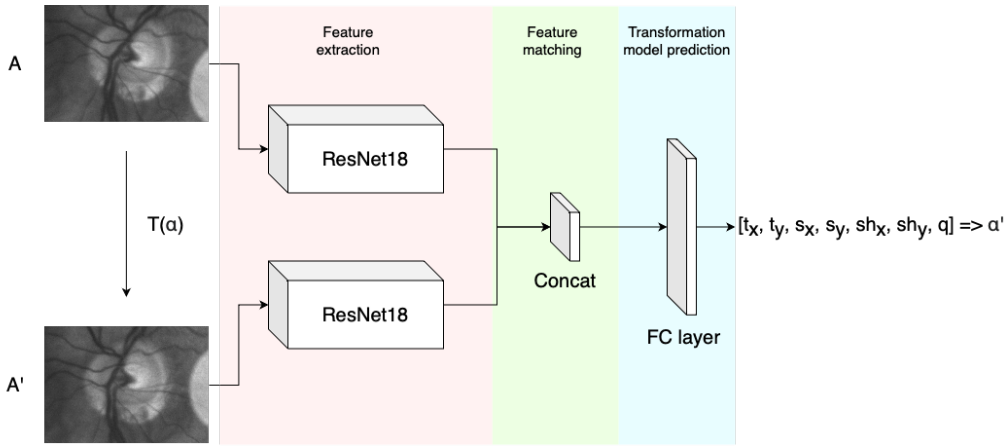


Fig. 4.3: Architecture of used siamese network. On the left there is transformation of original image A into A' using transformation T with α parameters. Each image enters one branch of ResNet, and the output from that branch is then concatenated and propagated to a fully connected layer that outputs a vector of particular geometric transformation which is then converted to parameters of affine transformation α' according to Equation 1.11

4.2.3 Optimizing feature matching

The first evolution of the initial architecture was the optimization of the feature matching step, which was performed by concatenation.

Subtraction

A literature search revealed a similarly simple and frequently used method of subtraction. This method was used for example in the WarpNet, which is network used for image matching, where it showed promising results [18]. The implementation of the subtraction only meant replacing the concatenation block which was very simple operation. Using subtraction instead of concatenation led to a noticeable reduction in network parameters, but only a slight improvement in results.

Correlation

Another popular approach for feature matching is the use of correlation, which measures the similarity between two feature maps. One of the early implementations of this method can be found in the work of Dosovitskiy *et al.* on FlowNet [10], where they introduced the concept of a correlation layer to compute the similarity between feature maps. This approach has been further developed and refined in various studies, such as Ilg *et al.* on FlowNet 2.0 [16] or Rocco *et al.* [26], where

they introduced a differentiable spatial to channel-wise correlation layer for efficient and effective feature matching.

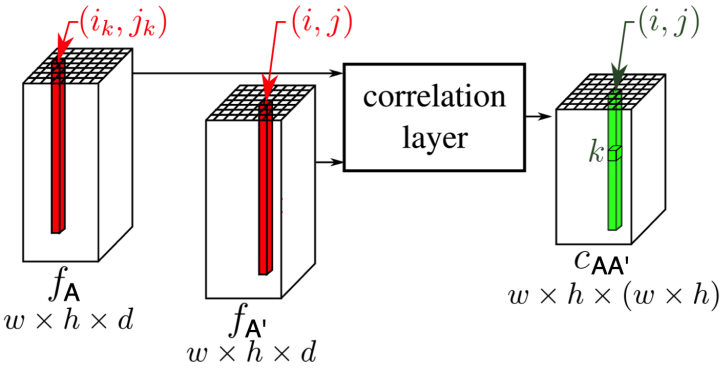


Fig. 4.4: Correlation layer computes correlation between individual features resulting in pairwise similarity map. Each spatial location of output map corresponds to spatial location each feature map. Taken and edited from [26].

The correlation layer calculates the dot product between feature vectors at a given spatial location in the reference feature map and all spatial locations in the target feature map. This results in a similarity map that represents the degree of correspondence between the two feature maps. The correlation layer can be implemented efficiently using convolutional operations, making it an attractive option for feature matching in deep learning architectures.

To incorporate the correlation layer into architecture, subtraction block was replaced with a correlation layer. This modification significantly increased the network's performance, outperforming both the concatenation and subtraction models. Moreover, the correlation layer introduced a higher level of robustness to the network, enabling it to better handle scale and rotation variations between the input images. The increased performance can be attributed to the layer's ability to effectively capture and exploit local similarities in the feature maps.

Despite the evident improvements, the correlation layer also increased the computational complexity of the model, as it required additional operations to calculate the similarity maps. However, the benefits in terms of performance and robustness justified the increased complexity, making the correlation-based model the preferred choice for this architecture.

4.2.4 Better regression

To further improve the performance of the network, the regression part of the architecture, which is responsible for predicting the geometric transformation parameters, was optimized. In the initial architecture, the regression block only consisted of a single fully connected layer. However, this simple design might not be sufficient to capture the complex relationships between the input features and the desired transformation parameters.

To address this limitation, a simple regression block consisting of a series of convolutional layers followed by a batch normalization and a ReLu activation function was implemented. After the last convolutional layer, a fully connected layer with an output dimension corresponding to the number of predicted parameters is added. Experimental results demonstrated that the new regression block led to significant improvements in the network's performance, as it was better able to capture the underlying relationships between the input features and the desired transformation parameters.

In conclusion, the introduction of a correlation layer and a more sophisticated regression block significantly improved the performance and accuracy of the network, which is verified in the next chapter of the evaluation. Final proposed architecture can be seen in Figure 4.5.

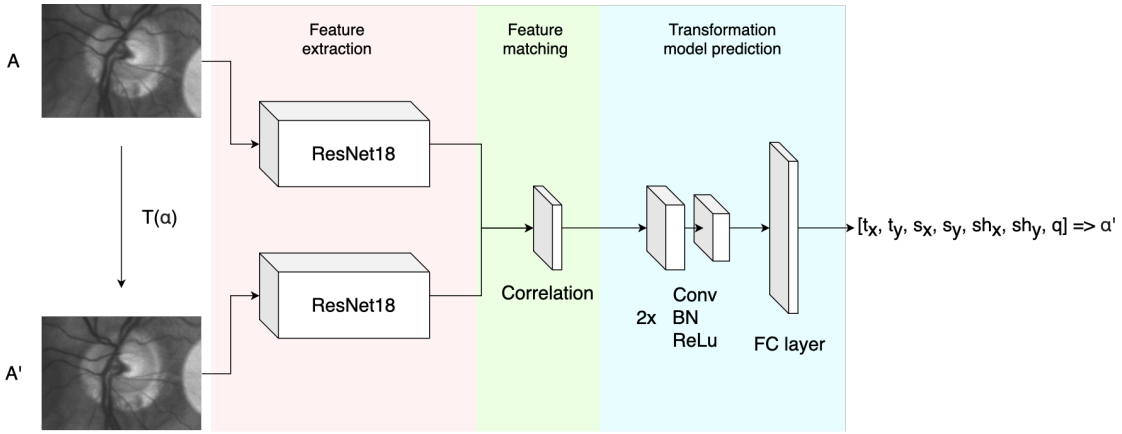


Fig. 4.5: Final architecture of used siamese network. On the left there is transformation of original image A into A' using transformation T with α parameters. Each image enters one branch of ResNet, and the output from that branch is then piped into correlation layer which results in pairwise similarity map. This map is then propagated into series of convolution layers, batch normalization and ReLu activation function. Final layer is fully connected layer that outputs a vector of particular geometric transformation.

4.2.5 Implementation details

The implementation was done in the Python programming language using PyTorch [25] deep learning library. All calculations and training took place in the cloud-based Jupyter notebook environment using Google Colab Pro. This environment offers the use of NVIDIA V100 or A100 graphics cards depending on availability.

Each architecture modification was trained several times until convergence. This usually occurred around epoch 15. Each epoch had 1938 iterations (62 patients $\times$ 500 frames / 16 batches). At the same time, each epoch was validated with 1000 iterations (16 patients $\times$ 500 images / 8 batches). At start of each iteration a random transformation parameters were generated and were used to warp the original image to get second input image with corresponding transformation parameters. Then both images were center cropped to make them smaller and at the same time to avoid the input of images with black edges - potentially caused by the transformation.

During training, mean square error was used as the criterion function and Adam [19] as the optimizer usually with learning rate of 10^{-4}.

5 Results and discussion

In this chapter all evolutions of the proposed architecture will be evaluated and discussed and then the resulting architecture will be compared with conventional methods.

The whole evaluation is divided into 3 sections. First, the accuracy of the individual predicted parameters is evaluated, then the accuracy of the registration is evaluated using whole frames and finally, an evaluation is performed on the provided video sequences where the focus is on the overall video stabilization.

5.1 Accuracy of predicted parameters

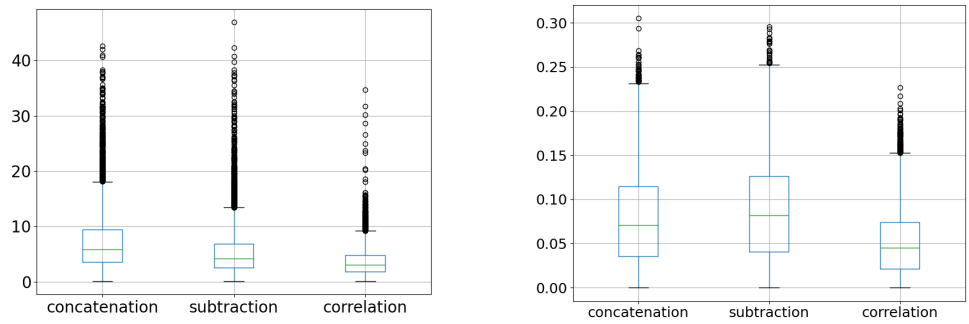
The accuracy of the predicted parameters was one of the main metrics used to optimize the overall architecture because it is very easy to calculate and gives a clear overview of how the performance will evolve. To evaluate prediction accuracy of individual parameters, Mean Absolute Error (MAE) was used. The resulting values can be seen in the following Tables 5.1 and 5.2, where they are shown with the standard deviation in brackets.

Table 5.1 presents a comparison of the accuracy between the individual parameters (i.e. translation, scale, shear, and rotation) depending on the feature matching method, before the introduction of the regression block described in Section 4.2.4. It can be observed that the correlation layer has the best performance for predicting all parameters, with the lowest MAE values. The individual error values were also plotted in boxplots in Figure 5.1, again divided according to individual parameters and depending on the feature matching method.

Tab. 5.1: Mean absolute error of the individual parameters (averaged over both axes) of the predicted geometric transformation before adding the regression block.

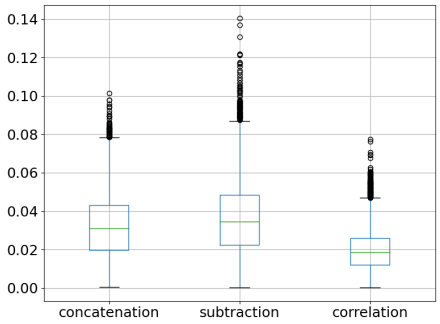
Matching layer	**translation**	**scale**	**shear**	**rotation**
concatenation	7.306 (5.527)	0.043 (0.022)	0.050 (0.034)	0.079 (0.053)
subtraction	5.420 (4.518)	0.040 (0.019)	0.048 (0.021)	0.086 (0.056)
correlation	3.630 (2.534)	0.020 (0.010)	0.039 (0.020)	0.051 (0.020)

Table 5.2 presents MAE values after introduction of the regression block. The addition of the regression block can be seen to improve the performance significantly across all three matching layers, reducing the MAE for each parameter. Once again, the correlation layer shows superior performance and is therefore the right choice

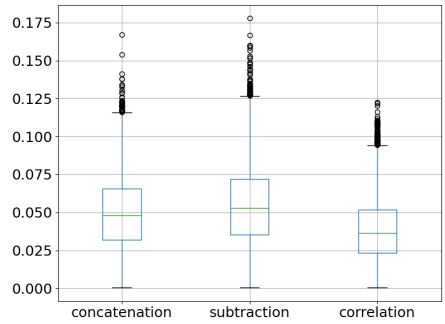


(a) translation

(b) rotation



(c) scale



(d) shear

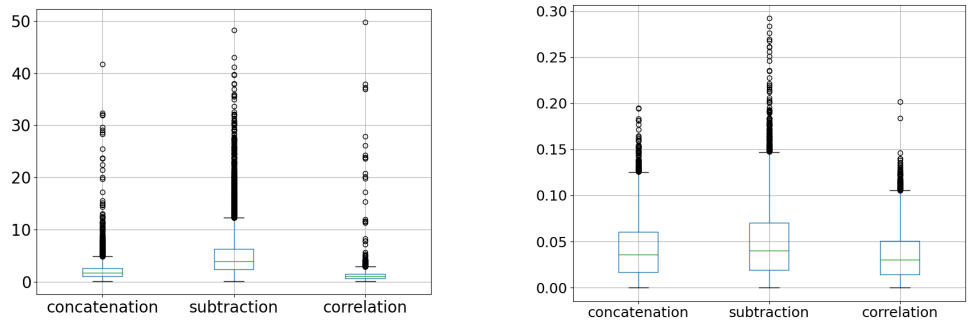
Fig. 5.1: This figure contains box plots of MAE values before the introduction of the regression block. It is divided by individual predicted parameters. Each box plot contains a box for each feature matching method.

for the final architecture. The individual error values for comparison after addition of the regression block were also plotted in boxplots in Figure 5.2.

Tab. 5.2: Mean absolute error of the individual parameters (averaged over both axes) of the predicted geometric transformation after regression block implementation.

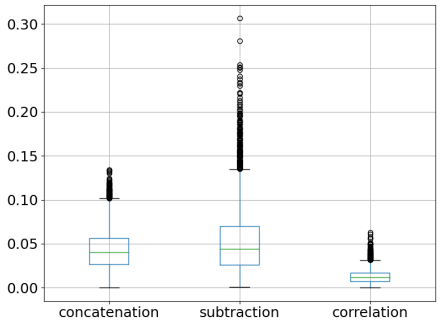
Matching layer	**translation**	**scale**	**shear**	**rotation**
concatenation	2.013 (1.815)	0.032 (0.017)	0.042 (0.026)	0.042 (0.030)
subtraction	3.087 (3.518)	0.036 (0.035)	0.047 (0.028)	0.049 (0.038)
correlation	1.179 (1.407)	0.012 (0.007)	0.034 (0.023)	0.035 (0.025)

One interesting issue appeared during the final evaluation. It was found that the accuracy of the prediction for the individual parameters that are separate for the x and y axis (i. e. translation, scale and shear) is noticeably different for each axis. This phenomenon is observable on all purposed architectures with very similar

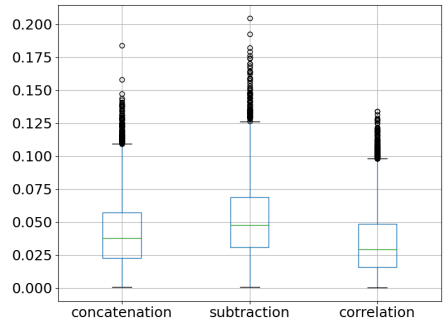


(a) translation

(b) rotation parameter



(c) scale



(d) shear

Fig. 5.2: This figure contains box plots of MAE values after the introduction of the regression block. It is divided by individual predicted parameters. Each box plot contains a box for each feature matching method.

deviations. Table 5.3 shows the percentage difference in the average error between the individual axes depending on the predicted parameter. All values are around 30 percent, which is quite a large deviation. This deviation is in favour of the x-axis, meaning that the accuracy of the parameters on the x-axis is better than on the y-axis.

The fact that this problem appears on all archictures means that there is some underlying problem in the archicture. When looking for where the problem might be, it was found that it is very likely caused by the input frames not having the same width and height. This dimension was specified at the very beginning without any thought. The width that corresponds to the x-axis is just 33.3 percent larger, which strongly suggests that this is the root cause.

Tab. 5.3: Percentage difference of the average error between the parameter axes, which are divided for each axis. The percentage difference is in favour of the x-axis.

	translation	**scale**	**shear**
dissimilarity	30.74 %	34.07 %	31.70 %

5.2 Registration evaluation using whole images

In this section the accuracy of the image registration will be evaluated by evaluating whole images. As in the previous case, the evaluation is performed on synthetically generated transformations, but here the error is calculated between the whole frames instead of the predicted parameters. Mean squared error (MSE) and structural similiraty (SSIM) were used for evaluation.

Table 5.4 shows the values for the individual metrics for each evolution of the proposed architecture. The table is divided into two columns depending on the before/after implmentation of the regression block. Again, all matching layers are compared to further support the results presented in the previous section. A similar trend can be seen here, i.e. that the correlation layer outperforms concatenation and subtraction and also that the implmentation of the regression block has significantly contributed to the overall performance of the network.

Tab. 5.4: Results of image registration using synthetic images. The table is divided into two columns before and after implmentation of the regression block. Each row shows metrics for a different matching layer.

	Single FC layer		**Regression block**	
Matching layer	**MSE**	**SSIM**	**MSE**	**SSIM**
concatenation	200.20 (134.94)	0.712 (0.118)	127.23 (92.83)	0.711 (0.116)
subtraction	192.99 (141.17)	0.708 (0.119)	170.40 (120.79)	0.714 (0.117)
correlation	125.04 (91.31)	0.701 (0.117)	54.22 (30.16)	0.699 (0.117)

More relevant results can be found in Table 5.5 where the final proposed architecture of the deep neural network (DNN) is compared with traditional registration methods based on feature matching. Two feature extraction methods were used here, namely SIFT (Scale-Invariant Feature Transform) and ORB (Oriented FAST and Rotated BRIEF). These features were then matched using the nearest neighbor method.

First, let us look at the first two rows of the table which show the values of the metrics calculated for the ground truth transformation and for the pair of images that are input to the registration process of all methods. The ground truth (GT) values should obviously be 0 for MSE and 1 for SSIM since it is a comparison of the original image and the back-warped image using known transformation parameters. But since the interpolation is performed twice, once when the transformed image is created and the second time when the back-transformation is performed, a certain error is introduced. The second row only shows the metrics between the original and warped image pair which is then input to the individual registration methods. So the goal is to get as close as possible to the ground truth values from the metrics of the input images.

As can be seen, the proposed architecture achieved very good results and outperformed both SIFT and ORB methods. These results indicate that the proposed DNN architecture is effective in accurately registering synthetic images compared to traditional methods based on feature matching. The proposed architecture shows superior performance in terms of minimizing the MSE metric and maximizing the SSIM metric, indicating better registration between the original and transformed images.

Tab. 5.5: Comparison of synthetic image registration results between the final deep neural network (DNN) architecture and traditional methods based on feature matching. The first two rows show the results for ground truth registration and for input images. The next rows show the results for the individual methods.

	MSE	**SSIM**
Ground truth	14.48 (9.48)	0.930 (0.052)
Input pair	429.36 (266.33)	0.623 (0.125)
Proposed DNN	54.22 (30.16)	0.722 (0.115)
SIFT	102.37 (238.05)	0.699 (0.117)
ORB	179.78 (614.63)	0.673 (0.106)

For a better understanding of these results, we can look at Figure 5.3 where the individual images are displayed together with the corresponding metrics. The first line shows the input images and the ideal ground truth (GT) registration. The bottom row shows a comparison between the proposed DNN architecture and traditional methods. In this case all methods achieved above average and very comparable results, which was not always the case.

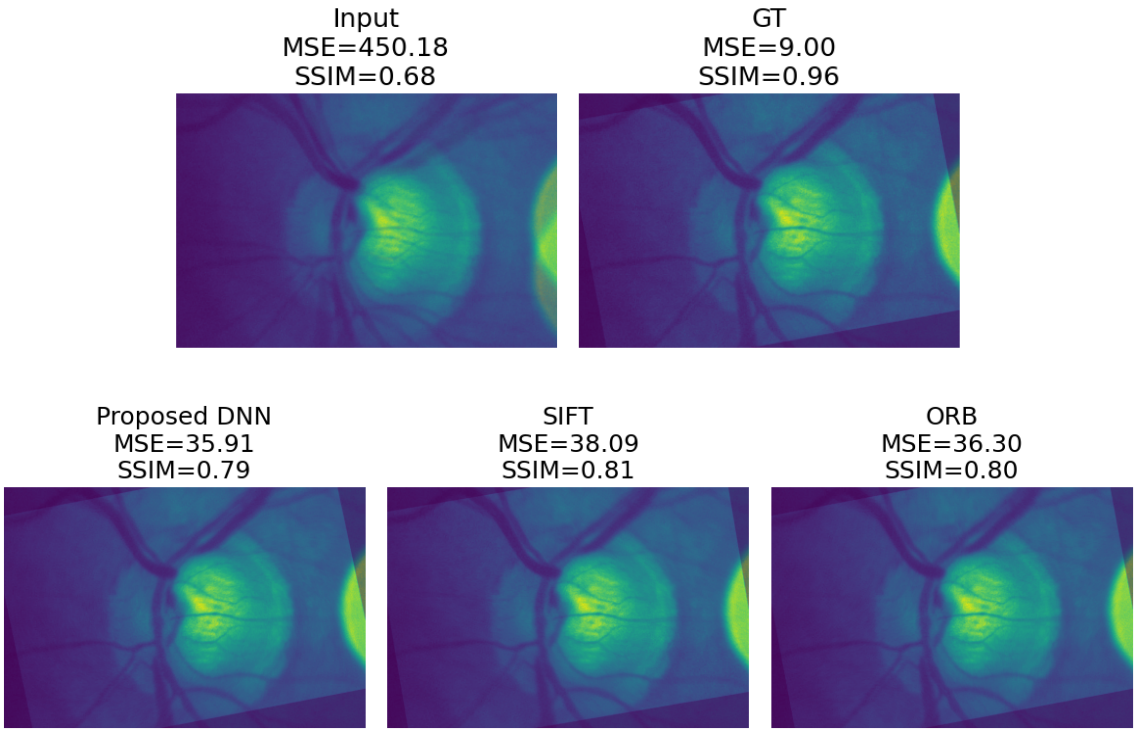


Fig. 5.3: Figure showing the registration evaluation of the individual used methods. The top row shows the input images and their ideal ground truth (GT) registration. The bottom row shows a comparison of the evaluation between individual methods.

Traditional methods (especially ORB) often had problems with poor quality (often blurred) or very dark images. Sometimes the problem was with feature matching where features were matched incorrectly (can be seen in Figure 5.4) resulting in very large MSE and low SSIM values. It even happened that they were not able to extract the features at all and thus were not matched, resulting in completely failed registration.

5.3 Video stabilization

In this section the registration of retinal images on real video sequences will be evaluated. For these video sequences and their successive frames, ground truth registrations are not available as in the case of the synthetic transformations generated by us in the previous sections. Again, mean square error (MSE) and structural similarity (SSIM) are used. The difference is that the individual metrics will not be calculated against the GT registered frame, but will be calculated against the first frame of the video sequence. The individual video sequences were registered to the

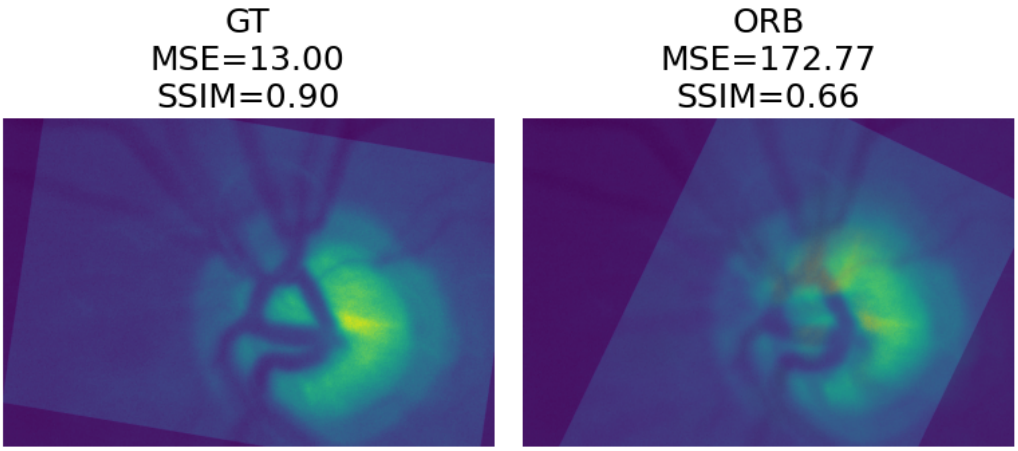


Fig. 5.4: An example of a case where the ORB method registered the input frames incorrectly due to false feature matches.

first frame, which means that the first frame is always fixed and each subsequent frame is moving. The result will be a sequence of individual metrics in time, where it will be possible to observe how the overall sequence changes and also the changes between individual frames.

For comparison with the proposed architecture, SIFT was used again, but ORB for its frequent problems described in the previous section was replaced by phase correlation, which on the other hand was not used because it cannot register according to all parameters of the affine transformation. Experimental results reveal that it is suitable for registration of real video sequences and achieves good results, which is convenient for comparison.

Tab. 5.6: Comparison of video sequences registration results between the final deep neural network (DNN) architecture and traditional methods. Individual values correspond to average values of individual frames averaged from all video sequences. The first row shows the values for the input video sequences before registration. The other rows show the values of the registered sequences for the individual methods.

	MSE	**SSIM**
Input sequence	337.77 (529.12)	0.594 (0.021)
Proposed DNN	186.73 (484.97)	0.702 (0.014)
Phase correlation	199.50 (336.73)	0.634 (0.019)
SIFT	122.75 (79.84)	0.654 (0.070)

Table 5.6 shows the average MSE and SSIM values of individual frames of the

video sequence against the first frame. The values are also averaged over all video sequences. The results show that all methods have somehow reduced the overall error, which means that the overall video has stabilized.

For a better interpretation of these results it is necessary to look at the individual video sequences separately. Figure 5.5 plots the MSE values of the individual frames of the video sequence against the first fixed frame. This plot shows a very good comparison of the registration of individual methods compared to the original unregistered video sequence. The original unregistered sequence (blue line) shows an upward trend which indicates an increasing difference from the first image. In the video this change is indicated by the scene gradually moving sideways.

In contrast, the lines of the registered sequences of the individual methods show more of a sideways trend, which means that the registered frames maintain a reasonably constant difference from the first frame. This is indicated in the video by the scene remaining in the same position and not moving sideways, which shows that all methods stabilize the video.

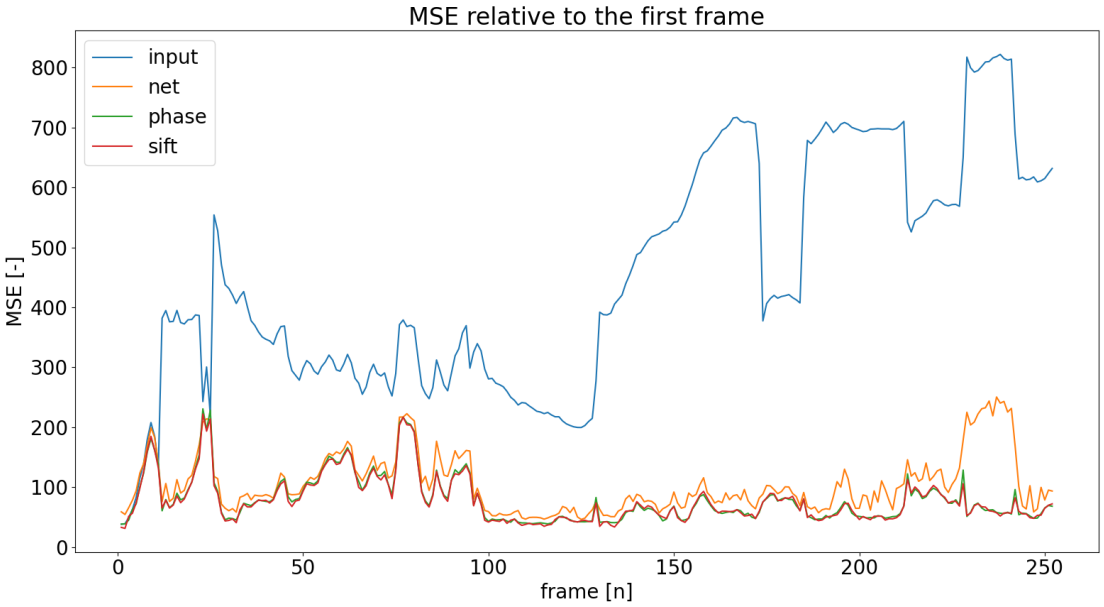


Fig. 5.5: MSE progression in example video sequence calculated relative to the first frame. Input line shows the progress of the original sequence, and the others show the progress of the registered sequences of individual method.

In Figure 5.6, the MSE values for the same video are plotted, but the MSE is calculated between consecutive frames, revealing rapid changes between frames most likely caused by rapid eye movement. These changes can be seen on the plot as significant peaks. For video to be properly registered it is necessary to remove

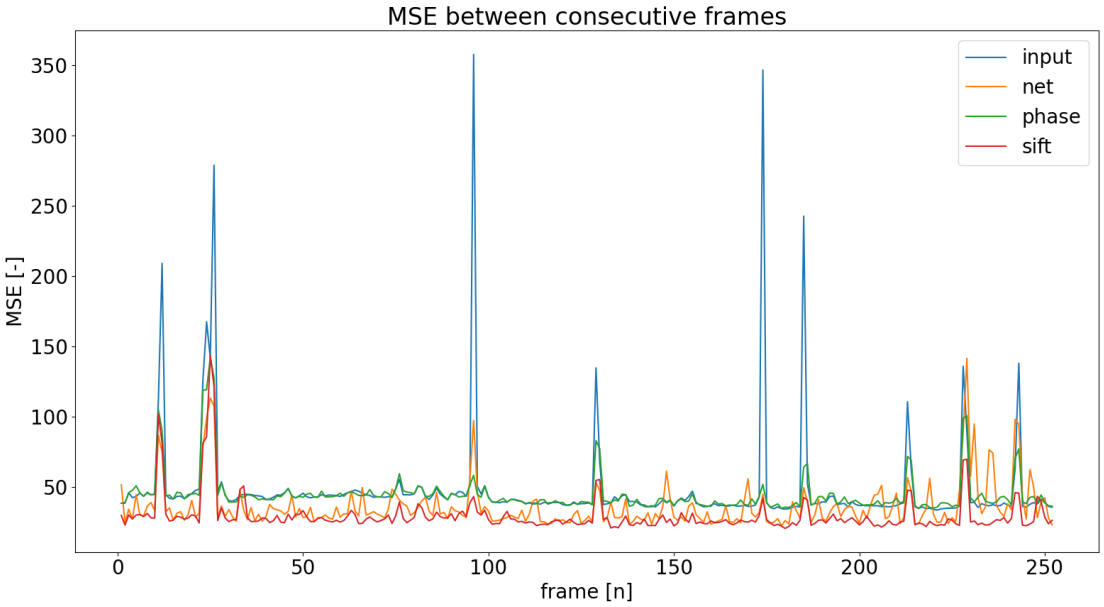


Fig. 5.6: MSE progression in example video sequence between consecutive frames. Input line shows the progress of the original sequence, and the others show the progress of the registered sequences of individual method.

these significant peaks, as they contribute to the largest error increments. This was often achieved, although the peak was not always completely removed.

If we look at the comparison of the individual methods both from the individual plots and from Table 5.7, where the average values of the metrics per example video are, the traditional methods have performed better. The main reason is that the proposed DNN architurce introduced new deviations into the registered video sequence. These deviations can be seen on both plots on the orange line, which is more spiky compared to the others. In the resulting registered video these deviations are clearly visible and are most often caused by rotation and shear.

This is the main problem of the proposed solution which was tried to remove from the beginning but unfortunately without any luck. This problem is related to the prediction of the individual parameters and the criterion function that was used in the training. If we look back at Table 5.2, we can see that the average absolute error of the translation parameter is two orders of magnitude larger than the error of the other parameters. This is simply due to the range in which the individual parameters vary. The translational parameters are in units or tens because they express translation in pixels, whereas rotation, for example, is in radians, which is usually in tenths and hundredths. When it comes to network training, the criterion function plays an important role. When calculating the error using the MSE cri-

Tab. 5.7: Comparison of single video sequence registration results. Individual values correspond to average values from all frames of one example video sequence. The first row shows the values for the input video sequence before registration. The other rows show the values of the registered sequence for the individual methods.

	MSE	**SSIM**
Input sequence	448.63 (192.26)	0.564 (0.020)
Proposed DNN	105.05 (48.20)	0.711 (0.013)
Phase correlation	78.88 (38.77)	0.672 (0.008)
SIFT	77.79 (37.95)	0.716 (0.028)

terion function, the parameters with larger values will logically have a larger error increment, which results in the network learning in this direction.

Several techniques have been tried to solve this problem, most of which consisted in training with the help of another criterion function. Among the function that have been tried are, Mean absolute percentage error (MAPE), which is an error function not scale-dependent, thus should be suitable for this task or weighted Mean squared error (wMSE), which is modification of MSE where the resulting error is weighted. These functions have been tried with different optimizers and their learning rates, but usually the resulting network had worse performance. Sometimes even convergence was not achieved.

To support the claim of learning in favor of translational parameters, we can look at Figure 5.7, where the MSE values for the registered frames are plotted, but only the translational parameters were used for registration using the proposed DNN architecture. It is clearly seen here that the architecture is on pair with traditional methods. First of all, it achieves smaller MSE values, which are as smooth as those of the traditional methods, and it is also evident in the resulting registered video sequence, where no errors in the form of rotation or shear are introduced. Even in terms of average MSE (73.60) and SSIM (0.731) values for this example video achieves better results.

Registrations of example video sequence for individual methods are available at `https://drive.google.com/drive/folders/1tfQ0_dI1KnqPJth5DfN8JS0hTE7fCRtg`

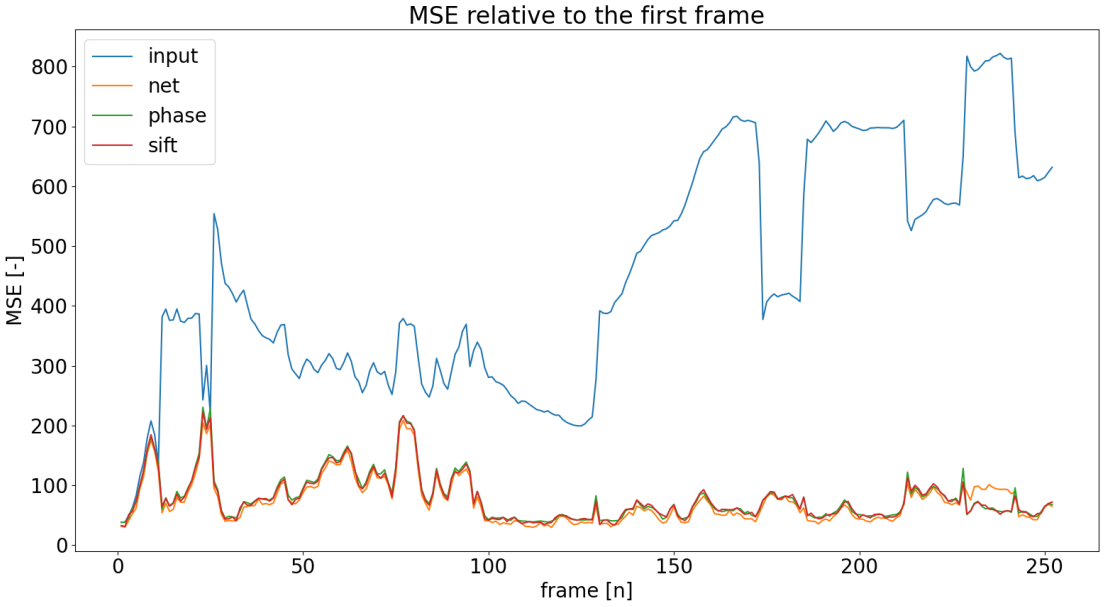


Fig. 5.7: MSE progression in example video sequence calculated relative to the first frame, where only translational parameters were used for registration from the proposed DNN architecture.

Conclusion

The subject of this thesis is registration of retinal images using deep learning. The aim was to get acquainted with this topic and to find a suitable method to implement in practice. In the theoretical part of the thesis the general principle of image registration is discussed and traditional methods are mentioned. Next, the methods of deep learning are discussed in detail, from which a suitable one is selected and implemented.

The chosen method is based on the prediction of the geometric transformation parameters. This method can take two frames as input and predict the geometric transformation parameters that are necessary to register these two frames. Since this is a method that requires supervised training, it was necessary to create synthetic images with known transformation parameters. The proposed architecture of deep neural network has a Siamese network architecture and its individual parts mimics the traditional image registration approach with the difference that it is able to predict the transformation parameters in one pass, which speeds up the whole process.

The entire architecture has undergone several optimizations which have resulted in significant performance improvements. At the end of the practical part, these optimizations were evaluated and the final architecture was compared with traditional image registration methods. The overall evaluation was divided into three parts, where each part evaluated the registration in a different way. In all parts, the proposed architecture was proven to be functional and suitable for retinal image registration. However, it needs to be further optimized and adapted to the given problem in order to completely replace traditional methods.

Bibliography

[1] FAST APPROXIMATE NEAREST NEIGHBORS WITH AUTOMATIC ALGORITHM CONFIGURATION. In *Proceedings of the Fourth International Conference on Computer Vision Theory and Applications*, SciTePress - Science and and Technology Publications, 2009, doi:10.5220/0001787803310340.
URL https://doi.org/10.5220/0001787803310340

[2] Bay, H.; Tuytelaars, T.; Gool, L. V.: SURF: Speeded Up Robust Features. In *Computer Vision – ECCV 2006*, Springer Berlin Heidelberg, 2006, s. 404–417, doi:10.1007/11744023_32.
URL https://doi.org/10.1007/11744023_32

[3] Boveiri, H. R.; Khayami, R.; Javidan, R.; aj.: Medical image registration using deep neural networks: A comprehensive review. *Computers & Electrical Engineering*, ročník 87, 2020: str. 106767, ISSN 0045-7906, doi: https://doi.org/10.1016/j.compeleceng.2020.106767.
URL https://www.sciencedirect.com/science/article/pii/S0045790620306224

[4] Chang, S.-H.; Cheng, F.-H.; Hsu, W.-H.; aj.: Fast algorithm for point pattern matching: Invariant to translations, rotations and scale changes. *Pattern Recognition*, ročník 30, č. 2, Únor 1997: s. 311–320, doi:10.1016/s0031-3203(96)00076-3.
URL https://doi.org/10.1016/s0031-3203(96)00076-3

[5] Chen, X.; Diaz-Pinto, A.; Ravikumar, N.; aj.: Deep learning in medical image registration. *Progress in Biomedical Engineering*, dec 2020, doi:10.1088/2516-1091/abd37c.
URL https://doi.org/10.1088/2516-1091/abd37c

[6] Czolbe, S.; Krause, O.; Feragen, A.: DeepSim: Semantic similarity metrics for learned image registration. *CoRR*, ročník abs/2011.05735, 2020, 2011.05735.
URL https://arxiv.org/abs/2011.05735

[7] Daudt, R. C.; Saux, B. L.; Boulch, A.: Fully Convolutional Siamese Networks for Change Detection. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, IEEE, Říjen 2018, doi:10.1109/icip.2018.8451652.
URL https://doi.org/10.1109/icip.2018.8451652

[8] DeTone, D.; Malisiewicz, T.; Rabinovich, A.: Deep Image Homography Estimation. 2016, doi:10.48550/ARXIV.1606.03798.
URL https://arxiv.org/abs/1606.03798

[9] Dey, S.; Dutta, A.; Toledo, J. I.; aj.: SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification. *CoRR*, ročník abs/1707.02131, 2017, 1707.02131.
URL http://arxiv.org/abs/1707.02131

[10] Dosovitskiy, A.; Fischer, P.; Ilg, E.; aj.: FlowNet: Learning Optical Flow with Convolutional Networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, IEEE, Prosinec 2015, doi:10.1109/iccv.2015.316.
URL https://doi.org/10.1109/iccv.2015.316

[11] Fischler, M. A.; Bolles, R. C.: Random sample consensus. *Communications of the ACM*, ročník 24, č. 6, červen 1981: s. 381–395, doi:10.1145/358669.358692.
URL https://doi.org/10.1145/358669.358692

[12] Fu, Y.; Lei, Y.; Wang, T.; aj.: Deep learning in medical image registration: a review. *Physics in Medicine Biology*, ročník 65, č. 20, Říjen 2020: str. 20TR01, doi:10.1088/1361-6560/ab843e.
URL https://doi.org/10.1088/1361-6560/ab843e

[13] Haskins, G.; Kruecker, J.; Kruger, U.; aj.: Learning deep similarity metric for 3D MR–TRUS image registration. *International Journal of Computer Assisted Radiology and Surgery*, ročník 14, č. 3, Říjen 2018: s. 417–425, doi:10.1007/s11548-018-1875-7.
URL https://doi.org/10.1007/s11548-018-1875-7

[14] He, A.; Luo, C.; Tian, X.; aj.: A Twofold Siamese Network for Real-Time Object Tracking. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, červen 2018, doi:10.1109/cvpr.2018.00508.
URL https://doi.org/10.1109/cvpr.2018.00508

[15] He, K.; Zhang, X.; Ren, S.; aj.: Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, červen 2016, doi:10.1109/cvpr.2016.90.
URL https://doi.org/10.1109/cvpr.2016.90

[16] Ilg, E.; Mayer, N.; Saikia, T.; aj.: FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, červenec 2017, doi:10.1109/cvpr.2017.179.
URL https://doi.org/10.1109/cvpr.2017.179

[17] Jan, J.: *Medical Image Processing, Reconstruction and Analysis: Concepts and Methods*, ročník 2. CRC Press, 2019.

[18] Kanazawa, A.; Jacobs, D. W.; Chandraker, M.: WarpNet: Weakly Supervised Matching for Single-View Reconstruction. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, červen 2016, doi: 10.1109/cvpr.2016.354.
URL https://doi.org/10.1109/cvpr.2016.354

[19] Kingma, D. P.; Ba, J.: Adam: A Method for Stochastic Optimization. 2014, doi:10.48550/ARXIV.1412.6980.
URL https://arxiv.org/abs/1412.6980

[20] Lee, M. C. H.; Oktay, O.; Schuh, A.; aj.: Image-and-Spatial Transformer Networks for Structure-Guided Image Registration. *CoRR*, ročník abs/1907.09200, 2019, 1907.09200.
URL http://arxiv.org/abs/1907.09200

[21] Lindeberg, T.: Scale Invariant Feature Transform. *Scholarpedia*, ročník 7, č. 5, 2012: str. 10491, doi:10.4249/scholarpedia.10491.
URL https://doi.org/10.4249/scholarpedia.10491

[22] Maes, F.; Vandermeulen, D.; Suetens, P.: Medical image registration using mutual information. *Proceedings of the IEEE*, ročník 91, č. 10, Říjen 2003: s. 1699–1722, doi:10.1109/jproc.2003.817864.
URL https://doi.org/10.1109/jproc.2003.817864

[23] Neculoiu, P.; Versteegh, M.; Rotaru, M.: Learning Text Similarity with Siamese Recurrent Networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, Association for Computational Linguistics, 2016, doi:10.18653/v1/w16-1617.
URL https://doi.org/10.18653/v1/w16-1617

[24] Noble, J. A.: Finding corners. *Image and Vision Computing*, ročník 6, č. 2, Kvìten 1988: s. 121–128, doi:10.1016/0262-8856(88)90007-8.
URL https://doi.org/10.1016/0262-8856(88)90007-8

[25] Paszke, A.; Gross, S.; Massa, F.; aj.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, editace H. Wallach; H. Larochelle; A. Beygelzimer; F. d'Alché Buc; E. Fox; R. Garnett, Curran Associates, Inc., 2019, s. 8024–8035.
URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-per
pdf

[26] Rocco, I.; Arandjelovic, R.; Sivic, J.: Convolutional Neural Network Architecture for Geometric Matching. *IEEE Transactions on Pattern Analysis and*

Machine Intelligence, ročník 41, č. 11, 2019-11-1: s. 2553–2567, ISSN 0162-8828, doi:10.1109/TPAMI.2018.2865351.
URL https://ieeexplore.ieee.org/document/8434328/

[27] Schroff, F.; Kalenichenko, D.; Philbin, J.: FaceNet: A Unified Embedding for Face Recognition and Clustering. *CoRR*, ročník abs/1503.03832, 2015, 1503.03832.
URL http://arxiv.org/abs/1503.03832

[28] Wu, G.; Kim, M.; Wang, Q.; aj.: Unsupervised Deep Feature Learning for Deformable Registration of MR Brain Images. In *Advanced Information Systems Engineering*, Springer Berlin Heidelberg, 2013, s. 649–656, doi:10.1007/978-3-642-40763-5_80.
URL https://doi.org/10.1007/978-3-642-40763-5_80

[29] Yang, X.; Kwitt, R.; Styner, M.; aj.: Quicksilver: Fast predictive image registration – A deep learning approach. *NeuroImage*, ročník 158, 2017: s. 378–396, ISSN 1053-8119, doi:https://doi.org/10.1016/j.neuroimage.2017.07.008.
URL https://www.sciencedirect.com/science/article/pii/S1053811917305761

[30] Zitová, B.; Flusser, J.: Image registration methods: a survey. *Image and Vision Computing*, ročník 21, č. 11, 2003: s. 977–1000, ISSN 0262-8856, doi:https://doi.org/10.1016/S0262-8856(03)00137-9.
URL https://www.sciencedirect.com/science/article/pii/S0262885603001379